HOCHSCHULE
MITTWEIDA
UNIVERSITY OF
APPLIED SCIENCES

BACHELORARBEIT

Herr
Klaus Walter

Untersuchungen zur Optimierung eines Verfahrens zur cookielosen Identifizierung von Webseitenbesuchern

2012

Fakultät **Mathematik** / **Naturwissenschaften** / **Informatik**

BACHELORARBEIT

Untersuchungen zur Optimierung eines Verfahrens zur cookielosen Identifizierung von Webseitenbesuchern

Autor:
Klaus Walter

Studiengang:
Informatik

Seminargruppe:
IF09w1

Erstprüfer:
Prof. Dr. Andreas Ittner

Zweitprüfer:
Dipl. -Ing.(FH) Michael Meisel

Mittweida, August 2012

Bibliografische Angaben

Walter, Klaus: Untersuchungen zur Optimierung eines Verfahrens zur cookielosen Identifizierung von Webseitenbesuchern, 60 Seiten, 32 Abbildungen, Hochschule Mittweida (FH), Fakultät Mathematik / Naturwissenschaften / Informatik

Bachelorarbeit, 2012

Referat

Die vorliegende Arbeit beschäftigt sich mit Untersuchungen zur Optimierung eines Verfahrens zur cookielosen Identifizierung von Webseitenbesuchern. Es wird ein Evaluierungsverfahren festgelegt, mit welchem das aktuell implementierte Verfahren untersucht und bewertet wird. Verschiedene Optimierungen werden umgesetzt und evaluiert.

I. Inhaltsverzeichnis

II. Abbildungsverzeichnis

III. Tabellenverzeichnis

1 Einleitung

1.1 Motivation

Am 08. Dezember 2011 wurde von einem beratenden Gremium der europäischen Kommission eine Empfehlung erstellt, wie die Direktive 2009/136/EC, eine Richtlinie, welche sich hauptsächlich auf den Computer-Datenschutz und insbesondere auf Datenschutz im Internet bezieht, in Bezug auf Online-Werbeverhalten umzusetzen ist. Bei dieser Empfehlung wird vor allem die Opt-in-Pflicht betont, welche eine ausdrückliche Erlaubnis zur Speicherung von Tracking-Cookies auf Client-PCs voraussetzt. Diese Richtlinie ist zwar in vielen Ländern noch nicht umgesetzt, zeigt aber, dass der Datenschutz bezüglich des Trackings mit Cookies erhöht werden wird. Des Weiteren wurde vor allem durch die Medien die Sensibilität im Bereich des Datenschutzes erhöht. Dies führt inzwischen zu einer nicht unerheblichen Anzahl von Nutzern, welche regelmäßig Cookies löschen [vgl. 12]. Die Quintessenz ist, dass zwar der Datenschutz der Nutzer in hohem Maß gesteigert wird, dies aber auch zu erheblichen Problemen führt. Trackingsysteme basieren hauptsächlich auf Cookie-Techniken. Diese werden sowohl im Bereich der Betrugserkennung als auch zu Marketing- sowie impliziten Personalisierungszwecken eingesetzt. Dies wird zukünftig verstärkt zur Anwendung von cookielosen Identifizierungs- oder Hybridverfahren führen. Vor allem im Bereich der Betrugserkennung stellt die Löschung von Cookies ein immenses Problem dar, welches durch Techniken wie Evercookie nur kurzfristig gelöst werden kann. Aus diesem Grund wurde im Rahmen der Masterarbeit von Herrn Fabian Seifert ein nicht auf Cookies basierendes Identifizierungsverfahren programmiertechnisch umgesetzt [vgl. 15]. Aufgrund der zunehmenden Notwendigkeit für ein cookieloses Identifizierungsverfahren soll verstärkt in diesem Bereich geforscht werden. Durch den Produktiveinsatz konnte evaluiert werden, dass das Identifizierungsverfahren noch einige Schwächen aufweist. Aus diesem Grund soll die Wiedererkennungsrate festgestellt und Möglichkeiten untersucht werden, diese zu verbessern.

1.2 Kapitelübersicht

Diese Arbeit besteht aus sieben Kapiteln. Im zweiten Kapitel werden die Grundlagen gelegt, um die nachfolgenden Ausführungen besser nachvollziehen zu können. Neben dem für das Thema spezifischen Wissen werden Kenntnisse über Cookie-basierte Wiedererkennungsverfahren, informationstheoretische Grundlagen sowie statistisches Grundwissen vermittelt. Das dritte Kapitel zeigt den Weg auf, der verfolgt wurde, um zu den in dieser Arbeit gewonnenen Erkenntnissen zu gelangen. In Kapitel 4 wird das aktuelle Verfahren betrachtet und bewertet. Das vorletzte Kapitel (5) behandelt die theoretischen Ansätze, um das aktuelle Verfahren verbessern zu können. Die Umsetzung dieser Ansätze und die Auswertung der darauf folgenden Optimierungen werden in Kapitel 6 behandelt. Im siebten und letzten Kapitel werden die gewonnenen Erkenntnisse zusammengefasst und es wird ein Ausblick gegeben, welche Richtungen zukünftige, darauf aufbauende wissenschaftliche Arbeiten einschlagen könnten sowie ein Fazit gezogen.

2 Grundlagen

2.1 Fingerprint

Die Electronic Frontier Foundation (EFF) ist eine nichtstaatliche Organisation, welche sich mit den Bürgerrechten im Internet beschäftigt. Im Jahr 2010 veröffentlichte diese ein Dokument mit dem Titel „How Unique Is Your Web Browser?“ [vgl. 5], in welchem die Resultate ihrer Plattform panopticlick.eff.org vorgestellt wurden. Diese Plattform stellt die Umsetzung eines Fingerprint-Algorithmus dar. Damit sollte gezeigt werden, wie effektiv die Wiedererkennung anhand der Gerätekonfiguration ist, was wiederum von zahlreichen kommerziellen Produkten zum damaligen Zeitpunkt schon genutzt wurde. Die dafür benötigten Konfigurationsdaten können beim Besuch einer Webseite vom Betreiber aus verschiedensten Quellen gewonnen werden.

Daten	Quelle
IP	IP-Header
Accept-Headers	HTTP-Header
Useragent	HTTP-Header
Cookies erlaubt	HTTP-Header
Bildschirmauflösung	AJAX
Zeitzone	AJAX
Plugins	AJAX
System-Schriftarten	AJAX
Mime-Types	AJAX
Super-Cookies	clientseitige Scriptsprachen
Aufenthaltsort	Geolocation (IP)

Tabelle 2.1: Daten & Quellen der Konfigurationsdaten

Für die Plug-In- und Schriftartenerkennung gibt es bereits fertige Softwarelösungen. Beispiele hierfür sind die JavaScript-basierte Software PluginDetect [vgl. 7] und FontDetect [vgl. 9].

Die Untersuchungen der EFF von 470.161 Fingerprints ergaben, dass 83,6 % dieser einzigartig waren, 8,2 % sich in einer Menge von 2-9 und 8,1 % einer Menge über 10 Geräten zuordnen ließen [vgl. 5, S. 2]. Ein großes Problem stellt die hohe Instabilität der Abdrücke dar. Hierbei wiesen 37,4 % der Geräte/User, deren letzter Besuch 24 Stunden zurücklag und die Cookies zugelassen hatten, beim nächsten Besuch einen veränderten Abdruck auf. Die hohe Anzahl von Merkmalen der Gerätekonfigurationen (Schriftarten, Plugins, etc.) mindert dieses Problem [vgl. 5, S. 11]. Zu beachten ist hierbei die Tatsache, dass Nutzer von „Panopticlick“ sehr datenschutzaffin sind und die Ergebnisse

auf anderen Plattformen abweichen dürften.

Besonderes Augenmerk bei der Optimierung sollte also auf der Wiedererkennung trotz geringfügiger Veränderungen der Konfiguration liegen. Von besonderer Bedeutung hierbei dürften Konfigurationsmerkmale sein, welche einer niedrigen Änderungsrate unterliegen und einen hohen Unterscheidungswert hinsichtlich ihrer Clients aufweisen. Ein Beispiel hierfür wäre eine per Geolokation ermittelte Ortsangabe. Diese unterliegen bei einer Auflösung bis auf Stadtebene einem hohen Unterscheidungswert und einer niedrigen Änderungsrate. In dieser Arbeit wird mehr Wert auf die Unterscheidung zwischen den Clients gelegt, aber durch eine Einbeziehung einer Änderungsrate, zum Beispiel in die Gewichtung, könnte durchaus eine Verbesserung der Wiedererkennung erzielt werden. Zu beachten ist auch, dass es Merkmale geben kann, die immer oder sehr oft zusammen auftreten. Diese gilt es zu ermitteln und Maßnahmen zu ergreifen (Entfernung von Attributen oder eine niedrigere Gewichtung), um eine erhöhte Gewichtung zu vermeiden.

2.2 Informationstheoretische Grundlagen

Um eine Gewichtung der einzelnen Merkmale (z. B. Java 1.6) eines Vektors (Fingerprints) aufgrund des Differenzierungsgrads des zugehörigen Clients zu anderen Clients verstehen zu können, müssen zuerst informationstheoretische Grundlagen [vgl. 18] geschaffen werden. Hierfür werden zuerst der Informationsgehalt, darauf aufbauend verschiedene Entropiemaße und schlussendlich die Transinformation behandelt.

Gegeben seien zwei diskrete Zufallsvariablen X mit dem Wertebereich $x_1, x_2, \ldots \in R$ und Y mit dem Wertebereich $y_1, y_2, \ldots \in R$.

Informationsgehalt

Der Informationsgehalt nach Shannon, auch Überraschungswert genannt, wurde in seiner Arbeit „Mathematical Theory of Communication“ [vgl. 16] definiert. Es handelt sich um eine logarithmische Größe einer Information (z. B. eines Zeichens), welche einzig und allein von der Wahrscheinlichkeit des Auftretens dieser abhängt. Daraus ergibt sich die Folgerung, dass je höher die Wahrscheinlichkeit des Auftretens einer Information ist, der Informationsgehalt umso niedriger ausfällt. Daher ist der Informationsgehalt einer Information, die immer auftritt, auch 0.

Hauptzweck von Shannons Arbeit war zum damaligen Zeitpunkt, die optimale Codierung von Nachrichten zu finden und damit auch die dafür nötige Bandbreite zu bestimmen.

Sei $a \in N$ die Gesamtanzahl der Zeichen des Codierungsalphabets, dann gilt folgende

Gleichung für den Informationsgehalt:

$$I(x_i) = -log_a(p(x_i)) \tag{2.1}$$

Der Informationsgehalt des Gesamtinformationssystems (z. B. einer Nachricht in der Kommunikationstheorie) setzt sich aus der Summe der Informationsgehalte der einzelnen Informationen zusammen. Sollte die Basis des Logarithmus 2 betragen, wird es mit der Einheit Shannon bezeichnet.

Entropie

Bei der Entropie handelt es sich um die Summe der mit den Auftrittswahrscheinlichkeiten gewichteten Informationsgehalte. Sie trifft eine Aussage, wie gut die Informationen aufgrund ihrer Wahrscheinlichkeiten verteilt sind und ist somit ein Maß für die Unsicherheit einer Zufallsvariable (Merkmal) [vgl. 4, S. 13].

$$H(X) = \sum_{i=1}^{n} I(x_i) * p_(x_i) \tag{2.2}$$

Das beliebteste Beispiel für die Entropie ist das Werfen zweier Münzen. Beide weisen einen Informationsgehalt von 1 auf, da die Wahrscheinlichkeit des Auftretens von Kopf oder Zahl jeweils 50 % beträgt.

$$H = 0,5 * 1 + 0,5 * (-1) = 1 \tag{2.3}$$

Hier beträgt die Entropie das Maximum von 1 (bei zwei Ausprägungen), da die Wahrscheinlichkeiten optimal verteilt sind (0,5:0,5 und 0,5:0,5).

Bei der Berechnung der Entropie bei einer Wahrscheinlichkeit von 0 ist zu beachten, dass

$$H = -log_a(0) * 0 = 0 \tag{2.4}$$

ist.

Gemeinsame Entropie

Gibt die gemeinsame Entropie zweier Zufallsvariablen an [vgl. 4, S. 16].

$$H(X,Y) = -\sum_{x \in X} \sum_{y \in Y} p(x,y) * logp(x,y) \tag{2.5}$$

Bedingte Entropie

Die bedingte Entropie von X und Y ist die Unsicherheit (Entropie) über Y, die verbleibt, wenn X bereits bekannt ist [vgl. 4, S. 6, 17].

$$H(X|Y) = H(X,Y) - H(Y) \tag{2.6}$$

Transinformation

Die Transinformation ist die Reduktion der Unsicherheit (Entropie) von X aufgrund der Kenntnis von Y. Dies wird in einem späteren Kapitel dazu genutzt, um festzustellen, wie groß die Reduktion der Entropie nach Kenntnisnahme des jeweiligen Attributs ist [vgl. 4, S. 21].

$$I(X;Y) = H(X) - H(X|Y) \tag{2.7}$$

2.3 Statistische Grundlagen

Um die Verwendung einer Stichprobe, die Umstellung des Skalenniveaus, die Korrelation zweier Merkmale sowie die Ähnlichkeitsanalyse zweier Vektoren im weiteren Verlauf ausführen zu können, sind wenige statistische Grundlagen notwendig.

Stichprobe

In der Statistik wird ein kleiner repräsentativer Ausschnitt des Ganzen (Grundgesamtheit) Stichprobe genannt. Repräsentativ ist eine Stichprobe, wenn alle Merkmale mit denen der Grundgesamtheit übereinstimmen. Die höchste Repräsentativität wird bei einer zufälligen Auswahl aus der Grundgesamtheit erzielt. Vom Ergebnis der Auswertung der Stichprobe können diese dann auf die Grundgesamtheit übertragen werden. Die benötigte Mindestgröße hängt von der Zusammensetzung der Grundgesamtheit ab. Es

gilt: Je höher der prozentuale Anteil am Ganzen, desto hochwertiger ist die Aussagekraft einer Hypothese über die Stichprobe.

Skalenniveau

Das Skalenniveau, auch als Messniveau oder Skalendignität bezeichnet, stellt in der Statistik eine Kategorisierung von Werten dar. Man unterscheidet zwischen:

Typ	Beschreibung
Nominal	Rein qualitative Merkmalsausprägungen ohne natürliche Ordnung
Ordinal	Qualitative Merkmalsausprägungen mit natürlicher Ordnung
Metrisch	Merkmalsausprägungen, die in einer Zahl bestehen und eine Dimension und einen Nullpunkt besitzen.

Tabelle 2.2: Skalenniveau

Beispiele:

Geschlecht: männlich/weiblich $\Rightarrow$ nominal
Schulnoten: 1/2/3/4/5/6 $\Rightarrow$ ordinal
Einkommen in Euro: 2.334,45/3.333,67 $\Rightarrow$ metrisch

Korrelation

Korrelationsmaße dienen dazu, die statistische Abhängigkeit der Ausprägungen (Merkmale) zueinander festzustellen [vgl. 14, S. 55]. Diese können dazu verwendet werden, die Merkmalstruktur des Fingerprintverfahrens zu verändern oder eine Wichtung der Merkmale vorzunehmen.

Chi-Quadrat-Unabhängigkeitstest

Beim Chi-Quadrat-Unabhängigkeitstest wird aufgrund der empirischen und erwarteten Häufigkeit untersucht, ob zwei Merkmale, welche bei jeder Beobachtung erfasst werden können, eine statistische Abhängigkeit aufweisen. Dies impliziert aber nicht zwingend einen kausalen Zusammenhang. Dies wird hauptsächlich bei nominalen und ordinalen Variablen genutzt. Als Nullhypothese wird die Unabhängigkeit der Merkmale gewählt. Die Wahl fällt auf ein gemischtes Fehlermaß [vgl. 14, S. 61]. Dies wird in einem späteren Kapitel dazu genutzt, Mehrfachgewichtungen zu vermeiden. Zu beachten ist, dass keine erwartete Häufigkeit kleiner als fünf sein darf.

Seien $h^{(1)} = (h_1^{(1)}, \ldots h_n^{(1)})$ und $h^2 = (h_1^{(2)}, \ldots h_n^{(2)})$ die Häufigkeitsvektoren zweier Merkmale, dann wird χ^2 folgendermaßen berechnet:

$$\chi^2 = \frac{1}{n}\sum_{i}^{r}\sum_{j}^{s}\frac{n * h_{ij} - h_i^{(1)} * h_j^{(2)}}{h_i^{(1)} * h_j^{(2)}} \tag{2.8}$$

Normierung von Merkmalen

Um Merkmale von einem bestimmten Intervall zu einem anderen zu transformieren, werden Normierungsfunktionen eingesetzt.

Min-Max Normierung

Sei $x \in R$ der Wert, der normiert werden soll, $minR$ und $maxR \in R$ die Spanne, in der sich die normierten Werte bewegen dürfen und $min \in R$ und $max \in R$ die Spanne, in der sich die Werte vor der Normierung befinden, dann kann der normierte Wert folgendermaßen ermittelt werden:

$$normiert(x) = \frac{x - min}{max - min} * (maxR - minR) + minR \tag{2.9}$$

Proximitätsmaße

Proximitätsmaße lassen sich in Ähnlichkeits- und Distanzmaße klassifizieren. Es werden nur implementierte aufgeführt. Eine Übersicht weiterer Proximitätsmaße lässt sich unter [19] einsehen.

Ähnlichkeitsmaße

Bei Ähnlichkeitsmaßen [vgl. 8] werden zwei Vektoren hinsichtlich ihrer Richtung und ihrer Länge verglichen. Dabei legen die unterschiedlichen Verfahren unterschiedlich viel Wert auf diese. Die Verfahren Cosinus, Dice und Jaccard beruhen auf dem kanonischen Skalarprodukt, wobei es sich um eine Abbildung von $R^n \times R^n \rightarrow R$ handelt [vgl. 6, S. 274]. Je höher das Ergebnis ausfällt, desto höher ist der Grad der Ähnlichkeit. Die verschiedenen Eigenschaften und Verhaltensweisen werden in 5.1. , vor allem im Hinblick auf den Fingerprint-Service, intensiver betrachtet.

Für weitere Aussagen soll gelten, dass $x, y \in R^n$ zwei Fingerprints sind, welche durch n Attributausprägungen gekennzeichnet sind.

Kanonisches Skalarprodukt in R^n

$$sim(x,y) = \sum_{i=1}^{n}(x_i * y_i) \tag{2.10}$$

Cosinus

$$sim(x,y) = \frac{\sum_{i=1}^{n}(x_i * y_i)}{\sqrt{\sum_{i=1}^{n} x_i^2} * \sqrt{\sum_{i=1}^{n} y_i^2}} \tag{2.11}$$

Der Nenner sorgt hierbei für eine Normierung auf das Intervall von Null bis Eins.

Dice-Koeffizient

$$sim(x,y) = \frac{2 * \sum_{i=1}^{n}(x_i * y_i)}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i} \tag{2.12}$$

Jaccard-Koeffizient

$$sim(x,y) = \frac{\sum_{i=1}^{n}(x_i * y_i)}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i - \sum_{i=1}^{n}(x_i * y_i)} \tag{2.13}$$

Overlap-Koeffizient

$$sim(x,y) = \frac{\sum_{i=1}^{n} min(x_i, y_i)}{min(\sum_{i=1}^{n} x_i, \sum_{i=1}^{n} y_i)} \tag{2.14}$$

Pearson-Korrelationskoeffizient

Bei einer Division durch Null wird Null zurückgegeben.

$$kor(x,y) = \frac{n * (\sum_{i=1}^{n}(x_i * y_i)) - \sum_{i=1}^{n} x_i * \sum_{i=1}^{n} y_i}{\sqrt{[n * (\sum_{i=1}^{n} x_i^2) - (\sum_{i=1}^{n} x_i)^2]} * \sqrt{[n * (\sum_{i=1}^{n} y_i^2) - (\sum_{i=1}^{n} y_i)^2]}} \tag{2.15}$$

Die Ähnlichkeitsaussage wird im Intervall [-1,1] getroffen. Daher muss der Wert dementsprechend auf das Intervall [0,1] normiert werden.

Distanzmaße

Analog zu den Ähnlichkeitsmaßen wird hier auch aufgrund der Richtung und der Länge bewertet. Dabei wird die Distanz zwischen den Vektoren bestimmt. Je geringer diese ist, desto ähnlicher sind sich zwei Vektoren.

Für weitere Aussagen soll gelten, dass $x, y \in R^n$ zwei Fingerprints sind, welche durch n Attributausprägungen gekennzeichnet sind.

Euklid

$$distance(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.16}$$

Manhattan

$$distance(x,y) = \sum_{i=1}^{n}|x_i - y_i| \tag{2.17}$$

Bei einer binär nominalen Normierung ist der Manhattan-Koeffizient mit dem vom Euklid gleichzusetzen, erfordert allerdings weniger Rechenaufwand.

Transformation von Distanzen zu Ähnlichkeitswerten

Sei $x \in R$ eine Distanz und d_{max} die maximale Distanz, dann gibt es folgende Möglichkeiten, Distanzwerte in einen Ähnlichkeitswert umzuwandeln [vgl. 3]:

Linearkombination der Grenzbedingungen

Die maximale Distanz bei der folgenden Illustration beträgt 10.

$$f(x) = 1 - \frac{x}{d_{max}} \tag{2.18}$$

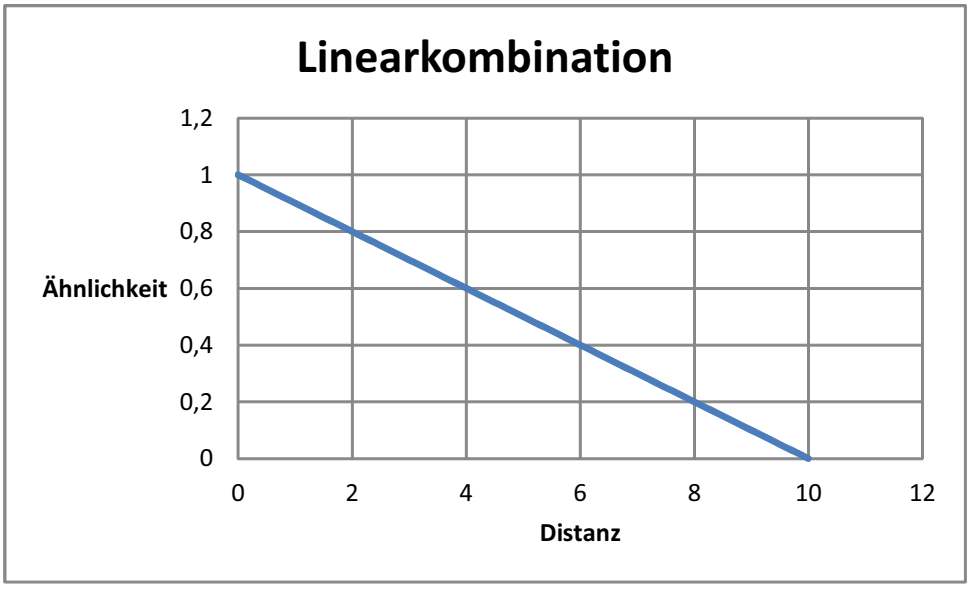

Abbildung 2.1: Linearkombination der Grenzbedingungen

Dynamische Sensibilität

$n \in N$ beträgt bei der folgenden Illustration 1. Bei n handelt es sich um einen frei wählbaren Parameter.

$$f(x) = e^{-x^n} \tag{2.19}$$

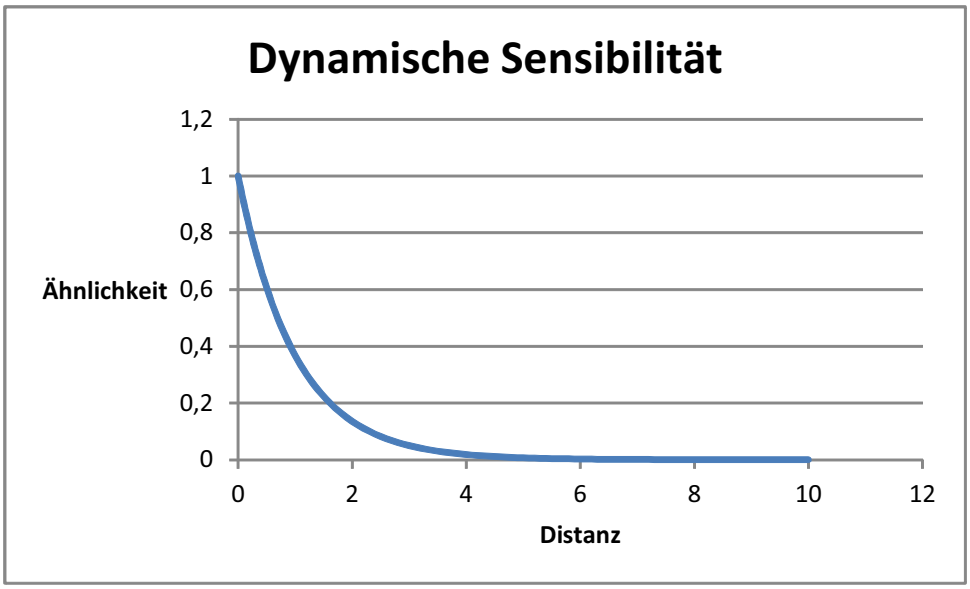

Abbildung 2.2: Dynamische Sensibilität

Parametrisierbare Funktion

Seien $t \in R$ die Distanzwerte bei 0 und $s \in R$ die globale Auswirkung, dann wird durch

$$f(x) = \frac{1}{1 + (\frac{x}{s})^t} \tag{2.20}$$

die parametrisierbare Funktion ausgedrückt.

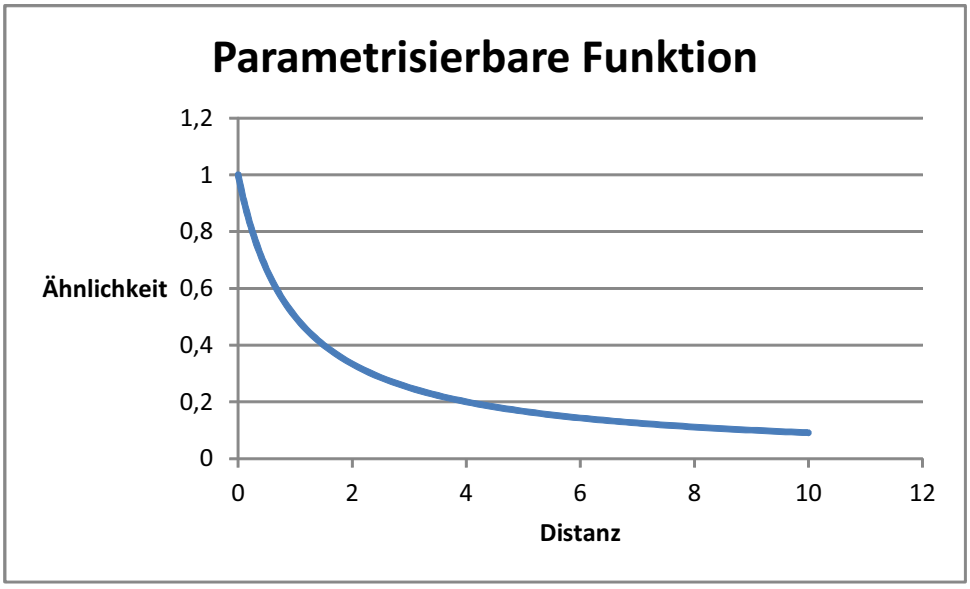

Abbildung 2.3: Parametrisierbare Funktion

2.4 Grundkenntnisse Data-Mining

Klassifikation

Unter diesem Begriff versteht man die Zuordnung von Elementen zu bereits vorhandenen Klassen. Die Zuordnungsentscheidung zu einer Klasse wird aufgrund der Merkmale der Elemente getroffen. Ein klassisches Beispiel für eine Klassifikation bieten zum Beispiel Kraftfahrzeugmodelle. Diese kann man in verschiedene Klassen wie Kompaktklasse, Kleinwagen, Mittelklasse und viele weitere unterteilen. Als Merkmale zur Klassifizierung dienen hierbei der Preis, die Fahrzeugmaße sowie die Anzahl der Sitze.

2.5 Cookiebasierte Wiedererkennungsverfahren

Bei Cookie-basierten Wiedererkennungsverfahren kann mit unterschiedlichen Arten von Cookies gearbeitet werden.

HTTP-Cookies

Hierbei handelt es sich um die Möglichkeit, ohne die technische Notwendigkeit der Nutzererlaubnis, per HTTP Cookies, d. h. kleinen Textdateien, Daten auf dem Client-PC speichern zu können.

Super-Cookies

Bei Super-Cookies handelt es sich um Speichermöglichkeiten, welche über Drittsoftware wie zum Beispiel Adobe Flash oder Silverlight zur Verfügung gestellt werden, um Informationen ohne explizite Nutzererlaubnis zu sichern. Großer Vorteil gegenüber HTTP-Cookies ist die browserübergreifende Speicherung.

Evercookie

Evercookies sind eine Kombination von mehreren Cookies, Supercookies sowie anderen Speichermöglichkeiten auf Clients, um nach der Entfernung von Bestandteilen, z. B. der HTTP-Cookies, diese wiederherstellen und somit die Persistenz des Evercookies sicherstellen zu können [vgl. 10].

2.6 Fingerprint-Service

Der Fingerprint-Service ist ein RESTful-Web Service, welcher für die persistente Speicherung von Fingerprint-Daten sowie die analytische Auswertung der Daten entwickelt wurde. Die Speicherung der Daten erfolgt in der NoSQL-Datenbank Cassandra [vgl. 1]. Der Service ist in zwei Hauptkomponenten unterteilt. Der Collector enthält den Code zur Speicherung der Fingerprints und das Wiedererkennungsverfahren. Mit der Komponente Analyzer lassen sich die Daten statistisch auswerten.

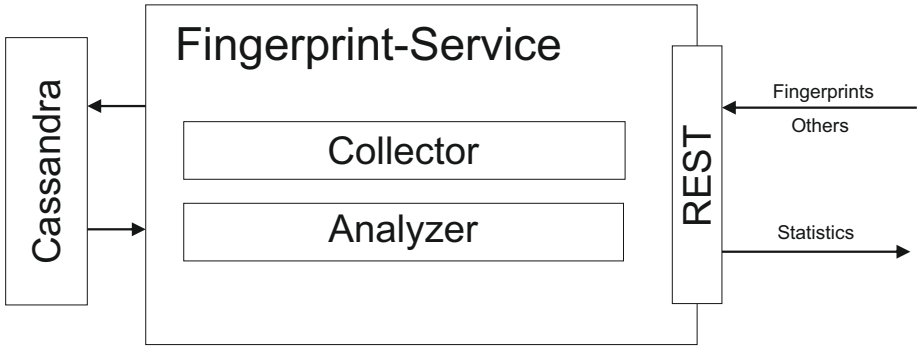

Abbildung 2.4: Fingerprint-Service

Visit

Bei einem Visit handelt es sich um einen Fingerprint, ergänzt um folgende Daten:

client
Enthält eindeutige ID, welche mit einem Cookie-basierten Wiedererkennungsverfahren gewonnen wurde. Im Fall der in dieser Arbeit verwendeten Daten beinhaltet dieses Feld Urchin Tracking Monitors (UTMs), welche mit Google Analytics generiert wurden.

tags
Dieses Feld enthält Tagginginformationen zur Klassifizierung des Visits. Dies können zum Beispiel eine bestimmte Sportmarke, eine User-ID oder Bestandteil einer selbstsprechenden URL sein.

ipAdress
Beinhaltet die zum Visit zugehörige IP-Adresse. Dabei kann es sich sowohl um IPv4 als auch IPv6 handeln.

refUrl
Enthält die URL des Referrers.

pageName
Name der Webseite, auf der die Daten des Visits aufgenommen wurden.

forwarded Header
Enthält X-Forwarded-For Header.

asserted client
Enthält den Key des Clients, dem der Visit zugewiesen wurde.

confidence
Resultat der Ähnlichkeitsanalyse mit dem Visit, der zur Klassifizierung des Visits geführt hat.

visitTime
Enthält das Datum und die Uhrzeit der Aufnahme des Visits.

cookieStatus
Aussage darüber, ob Cookies angenommen werden.

mappedID
ID eines anderen Visits, dessen Attribute mit denen dieses Visits übereinstimmen.

Attribut

Attribute sind Ausprägungen eines Visits. Dies können zum Beispiel die Browserversion, die Zeitzone oder eine installierte Schriftart sein.

Attributausprägung

Attributausprägungen sind Ausprägungen eines Attributs, d. h. die verschiedenen Werte, die ein Attribut annehmen kann. Um dies zu illustrieren, wird eine vereinfachte Darstellung eines Attributs Betriebssystem, welches in diesem Fall vier Ausprägungen hat, dargestellt.

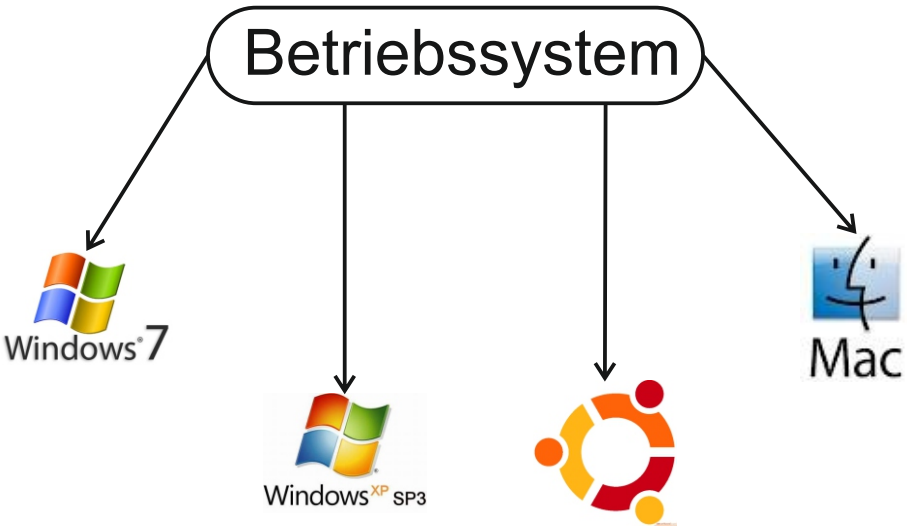

Abbildung 2.5: Attributausprägung

Column Families

Die Column Families, deren Spalten, Werte und Beschreibungen in Abbildung A.1 dargestellt sind.

Prozess des Wiedererkennungs-Verfahrens

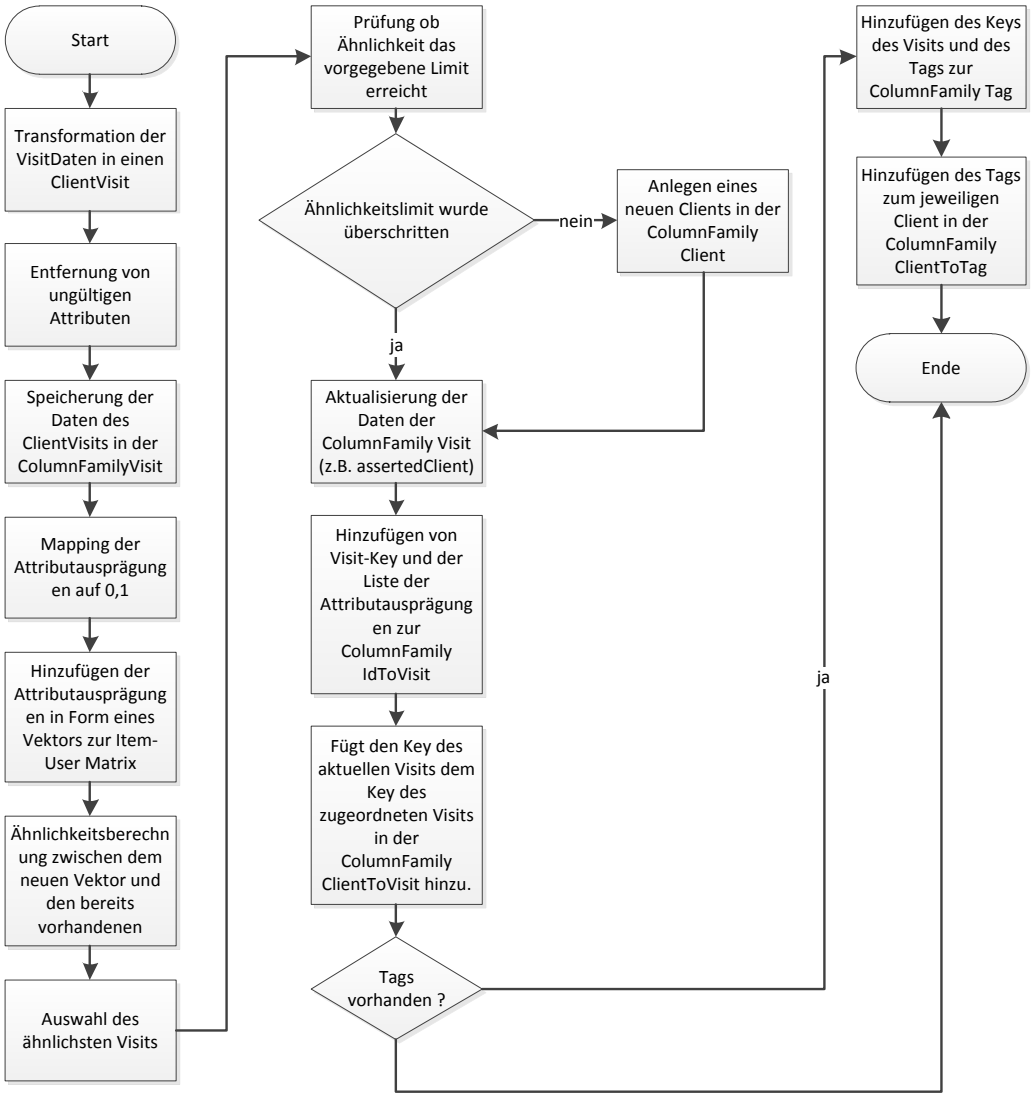

Abbildung 2.6: Prozess des Wiedererkennungs-Verfahrens

Aggregation

Aggregation bedeutet im Allgemeinen das Zusammenfassen von Etwas. Im Kontext dieser Arbeit soll dieser Begriff aber eindeutig definiert sein. Hierbei wird zur Illustration ein Beispiel einiger Fingerprints mit einer maximalen Anzahl von vier möglichen Attributausprägungen aufgezeigt.

Flash 11.3.300.268	Adobe Reader 10.010	Java 16.024	Java 15.019
0	1	1	0
1	1	1	0
0	1	1	0
0	1	1	0
1	1	1	1

Tabelle 2.3: Vektoren vor Aggregation

Es werden alle identischen Vektoren ermittelt und zu einem Vektor zusammengefasst.

Flash 11.3.300.268	Adobe Reader 10.010	Java 16.024	Java 15.019
0	1	1	0
1	1	1	0
1	1	1	1

Tabelle 2.4: Vektoren nach Aggregation

3 Vorgehensweise

Um das Verfahren hinsichtlich der Wiedererkennung zu optimieren, ist es notwendig, ein Evaluierungsverfahren zu entwickeln. Danach müssen Daten zuerst gewonnen oder, wie in diesem Fall, schon vorhandene Fingerprint-Daten verwendet werden. Der nächste Schritt ist, dass aktuelle Verfahren mit dem Evaluierungsverfahren zu bewerten. Aufgrund dieser Baseline können dann Optimierungen bewertet werden. Die Betrachtung des aktuellen Verfahrens, theoretische Überlegungen zu Optimierungen sowie die Umsetzung und Auswertung, wurde aufgrund ihrer Wichtigkeit für diese Arbeit in eigene Kapitel ausgelagert. Zur Umsetzung der aufgeführten Vorgänge wurde das Fingerprint-Evaluierungstool (FET) entworfen.

3.1 Gewinnung von Fingerprintdaten

Um die Evaluierungsgüte des Wiedererkennungverfahrens zu erfassen, müssen neben den Daten des Fingerprints auch eindeutige Identifizierungsschlüssel eines Cookie-basierten Wiedererkennungsverfahrens erhoben werden. Hierfür können die unter dem Abschnitt 2.4. aufgeführten, sich in der Qualität unterscheidenden, Cookie-basierten Identifizierungsverfahren eingesetzt werden. Es gibt aber kein Verfahren, bei welchem eine hundertprozentige Wiedererkennung gewährleistet werden kann, da die Festlegung, ob der Fingerprint dem Client richtig zugeordnet wurde, von dem oben erwähnten Verfahren abhängt und es dem Nutzer des Clients jederzeit möglich ist, sein System und damit die Cookies zu verändern.

Datenquelle

Zur Gewinnung von Fingerprintdaten sollte auch die Quelle sorgfältig nach bestimmten Kriterien ausgewählt werden. Diese sind

- die Anzahl der erhobenen Fingerprints pro Tag,
- die Art des Cookie-basierten Wiedererkennungsverfahrens zur Evaluierung sowie
- die Höhe der Wiederkehrrate der Benutzer (je höher desto besser).

Aufgrund der benötigten statistischen Grundgesamtheit und der geringen Zeitspanne werden vorhandene Fingerprint-Daten eines Web-Portals verwendet. Die Wiedererkennung, welche zur Evaluierung nötig ist, basiert auf sogenannten Google-Cookies (HTTP-Cookies).

Extraktion von Fingerprintdaten aus einem Online-Portal

Um die aus dem Online-Portal gewonnenen Daten verwenden zu können, mussten diese vorher aus Cassandra extrahiert werden. Allerdings wurde nicht wie üblich auf Snapshots gesetzt, sondern der komplette Cassandra-Ordner kopiert, in welchem alle für die Datenbank benötigten Daten vorzufinden sind. Dieser wurde dann in das Evaluierungssystem eingebracht.

Grundsätzliche Erkenntnisse

Unbrauchbare Daten

Datensätze, deren Felder „client“ oder „assertedClient“ keinen Wert beinhalten, können nicht verwendet werden. Bei einem fehlenden Wert „client“ (Cookie-Id), was zum Beispiel dadurch verursacht worden sein kann, dass bei diesem Gerät die Annahme von Cookies verweigert wird, kann die Fingerprintzuweisung zum Client nicht evaluiert werden. Bei einem fehlenden Wert „assertedClient“ wurde keine oder eine fehlerhafte Wiedererkennung durchgeführt. Von 507.235 Visits (ohne vorhergehende Aggregation) können 9.118 nicht verwendet werden.

Cookies

Es wurden 107.860 Cookies ermittelt.

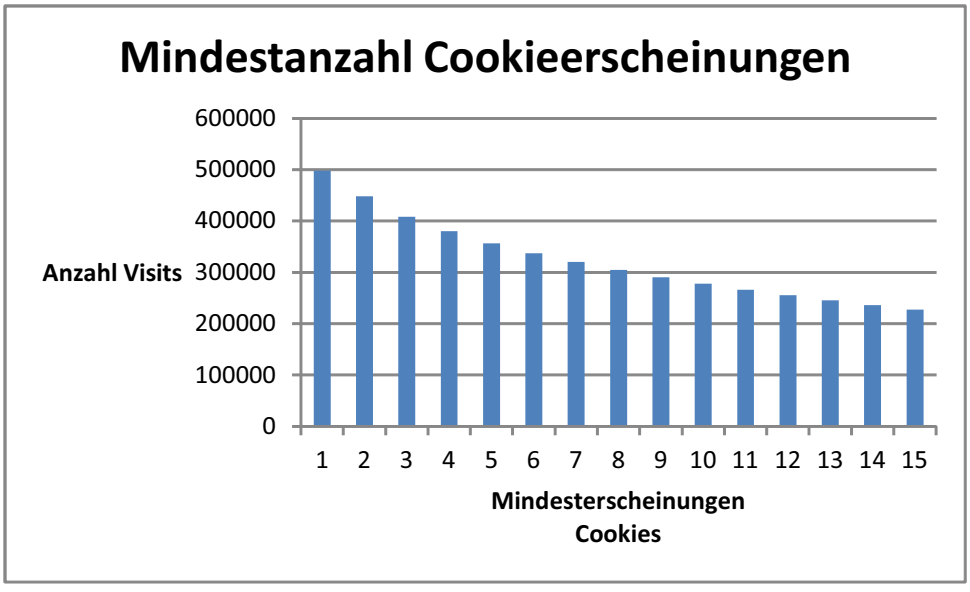

Abbildung 3.1: Mindestanzahl der Cookieerscheinungen

Das Diagramm zeigt deutlich, dass die meisten Nutzer des Portals dieses öfters als

fünfmal besuchen und damit optimale Bedingungen für die Optimierung des Verfahrens bieten. Durch die Häufigkeit der Besuche erhöht sich auch die Anzahl der veränderten Fingerprints eines Clients.

Attribute

Es wurden insgesamt 121 verschiedene Attribute mit ingesamt 12.981 Ausprägungen festgestellt. Die Bezeichnungen der Attribute und deren Anzahl von Attributausprägungen können den Tabellen B.1 - B.3 entnommen werden.

Länge der Visits

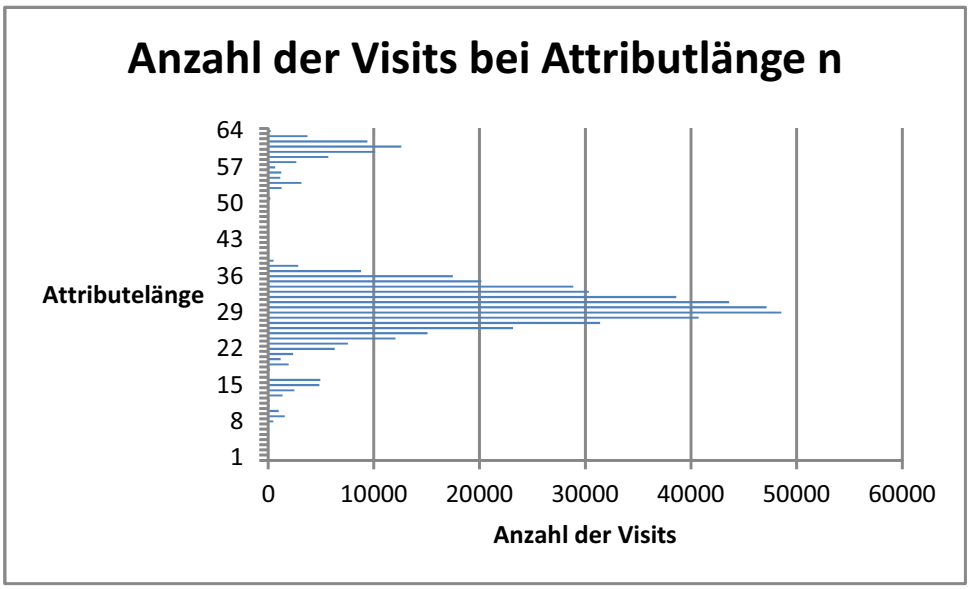

Abbildung 3.2: Anzahl der Visits bei Attributlänge n

Dieses Diagramm zeigt deutlich, dass die meisten Visits, welche erhoben werden, 18 bis 35 Attribute aufweisen. Drei weitere Häufungen zeigen sich bei einer Länge zwischen 7 und 9, 12 bis 15 sowie 52 und 63 Attributen. Eine Hypothese für sehr kurze Visits ist die Deaktivierung von JavaScript. Die hohe Anzahl von Attributen deutet darauf hin, dass sehr viele Schriftarten auf dem System installiert sind. Dies gilt es in zukünftigen Untersuchungen zu evaluieren.

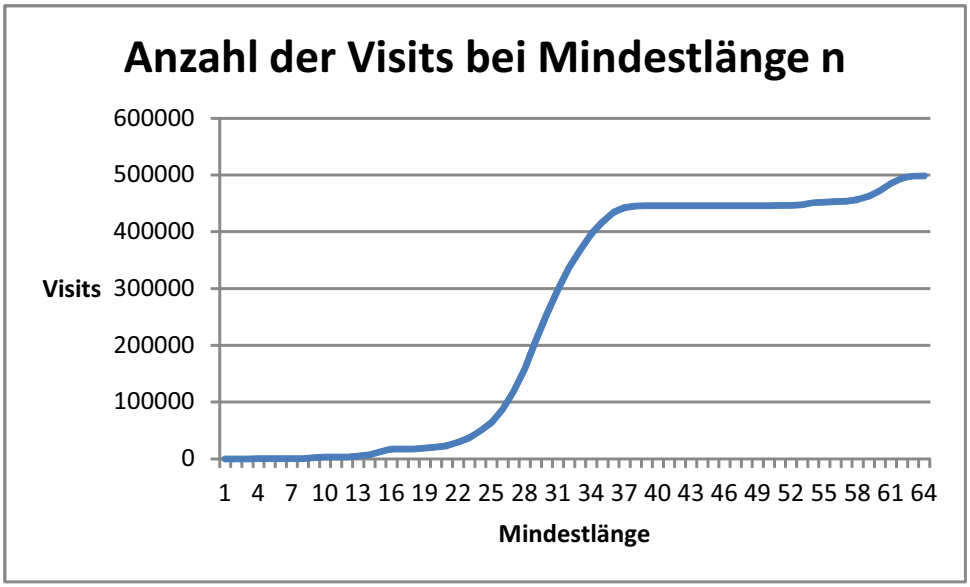

Abbildung 3.3: Anzahl der Visits bei Mindestlänge n der Attribute

3.2 Evaluierungsverfahren

Die Evaluierungsmetriken sollen eine Beurteilung der Wiedererkennungsrate eines Verfahrens sowie eine Vergleichsmöglichkeit zwischen den Verfahren ermöglichen. Eine Möglichkeit der Evaluierung wäre es, eindeutige angebotspezifische Identifizierungsstrings (z. B. Kundennummer) zu verwenden. Die dafür benötigten Loginvorgänge stellt aber nicht jedes Webangebot zur Verfügung. Außerdem wird eine bestimmte Teilmenge der Clients ausgeschlossen. Als Ergebnis erhält man eine nicht repräsentative Stichprobe. Da das Ergebnis einer Evaluierung nicht auf andere Plattformen übertragbar ist, wird diese Möglichkeit ausgeschlossen. Aus diesem Grund wird ein Cookie-basiertes Identifzierungsverfahren verwendet.

Terminologie des Evaluierungsverfahrens

Cookie-ID

Es handelt sich hierbei um einen eindeutigen Schlüssel, welcher von einem Cookie-basierten Wiedererkennungsverfahren erzeugt wird und einem Client zugeordnet ist.

assertedClient

AssertedClient ist ein eindeutiger Identifizierungsschlüssel eines Clients, welcher vom Fingerprint-Verfahren vergeben wird.

Gesamtanzahl Cookies

Hierbei wird nicht die gesamte Anzahl der Cookies, sondern die Gesamtanzahl der Cookies nach Entfernung ungültiger Visits und Überschreitung der Mindestanzahl von Visits einer Cookie-ID bezeichnet.

Metrik der Cookie-Map

Intention

Das Fingerprint-Verfahren wird aufgrund einer angenommenen, aber sehr unwahrscheinlichen Annahme evaluiert, dass das Cookie-basierte Identifizierungsverfahren zu hundert Prozent korrekt ist. Dies liegt darin begründet, dass es keine Möglichkeit gibt, eine insgesamt korrekte Evaluierung durchzuführen. Daraus lässt sich folgern, dass die Metriken gefunden werden müssen, welche die besten Evaluierungsmöglichkeiten bieten. Umgangssprachlich würde man dafür folgende Worte finden:

„Man muss nehmen, was man bekommt!“.

Sollten die Cookies gelöscht worden sein, wird laut dem Cookie-Identifizierungsverfahren angenommen, dass es sich um einen neuen Client handelt. Bei einer generellen Ablehnung von Cookies können die entsprechenden Visits nicht zur Evaluierung herangezogen werden. Dies sind beides Tatsachen, welche das Vertrauen in diese Metrik schmälern. Sie ist aber trotzdem ein verhältnismäßig guter Indikator für die Wiedererkennung des Fingerprint-Services. Ausgehend von einem zu hundert Prozent korrekten Verfahren, und wenn durch das Fingerprint-Verfahren alle Wiederkennungen ebenso korrekt durchgeführt würden, dann müsste nach dieser Annahme jeder Cookie-ID ein AssertedClient zugewiesen werden. Sollten einer Cookie-ID mehrere AssertedClients zugewiesen werden, so würde theoretisch eine falsche Wiedererkennung durch das Fingerprintverfahren durchgeführt. Dies soll durch die Metrik „DegreeofHardness“ ausgedrückt werden. Vor der Betrachtung dieser wird aber zuerst der Aufbau der Cookie-Map behandelt.

Cookie-Map

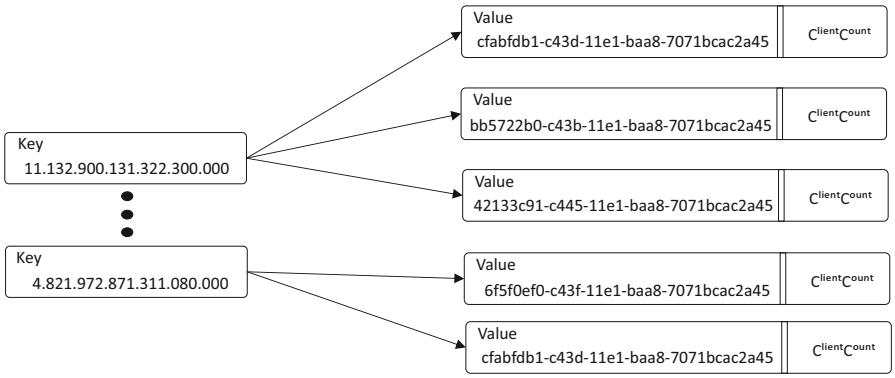

Abbildung 3.4: Cookie-Map

Bei der Cookie-Map handelt es sich um eine Hashmap, die als Key die Cookie-ID (String) und als Wert eine weitere Map nutzt, welche die UUID assertedClient und ein ClientCount-Objekt beinhaltet.

Die Klasse ClientCount ist folgendermaßen strukturiert:

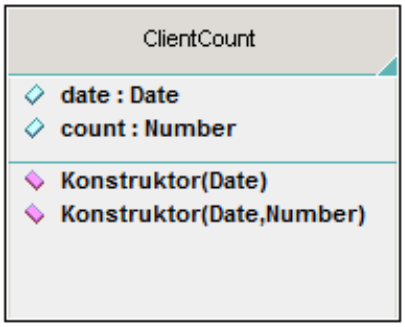

Abbildung 3.5: ClientCount

Besonderheit hierbei ist, dass auf die Klassenattribute direkt zugegriffen wird. Damit lassen sich die Anzahl der Visits pro AssertedClient per Cookie und das Datum der letzten Änderung auslesen. Untechnisch ausgedrückt werden jeder Cookie-ID die AssertedClients der Visits zugeordnet. Ein korrekter Cookie ist eine Zuordnung von einem AssertedClient zu einer Cookie-ID.

Abbildung 3.6: Metriken der Cookie-Map

degreeOfHardness (DOH)

Der DOH ist eine Metrik (Faktor), welche auf dem Verhältnis korrekter Cookies zur gesamten Anzahl der Cookies beruht und wird dementsprechend folgendermaßen berechnet:

$$degreeOfHardness = wrongCookiesCookieMap/GesamtanzahlCookies \quad (3.1)$$

Wenn der DOH sinkt, kann dies folgende Ursachen haben:

- Es ist eine erhöhte Löschung von Cookies erfolgt,
- die Wiedererkennung durch den Fingerprint-Service hat sich verbessert oder
- es ist eine Mischung von beidem.

Im Fall einer Steigung verhält es sich vice versa.

Metrik der Client-Map

Intention

Im Gegensatz zu der Metrik der Cookie-Map wird bei der Client-Map das Cookie-basierte Wiedererkennungsverfahren aufgrund des Fingerprintverfahrens evaluiert. Hierbei wird angenommen, dass die Güte der daraus resultierenden Metrik schlechter ist als der DOH, da die Vermutung nahe liegt, dass die Auswirkungen der Fehlerrate durch Löschung der Cookies niedriger ist als durch die Fehlerrate des Fingerprint-Verfahrens. Dennoch kann diese Metrik, wenn man diese mit den anderen Metriken vergleicht, ein Hinweis für die Löschrate von Cookies sein. Daraus resultiert ein wichtiger Indikator zur Einschätzung der Korrelationsmetriken und dem DOH.

Client-Map

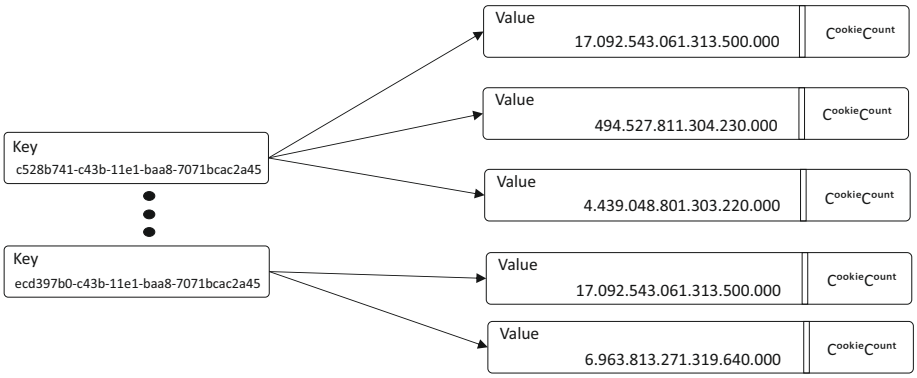

Abbildung 3.7: Client-Map

Im Gegensatz zur Cookie-Map werden hier ein Cookie und ein CookieCount-Objekt den AssertedClients zugewiesen. Das CountObjekt heißt bei der Client-Map CookieCount, verfügt aber über die gleiche Funktionalität wie die Klasse ClientCount. Ein korrekter Client ist eine Zuordnung von einer Cookie-ID zu einem AssertedClient.

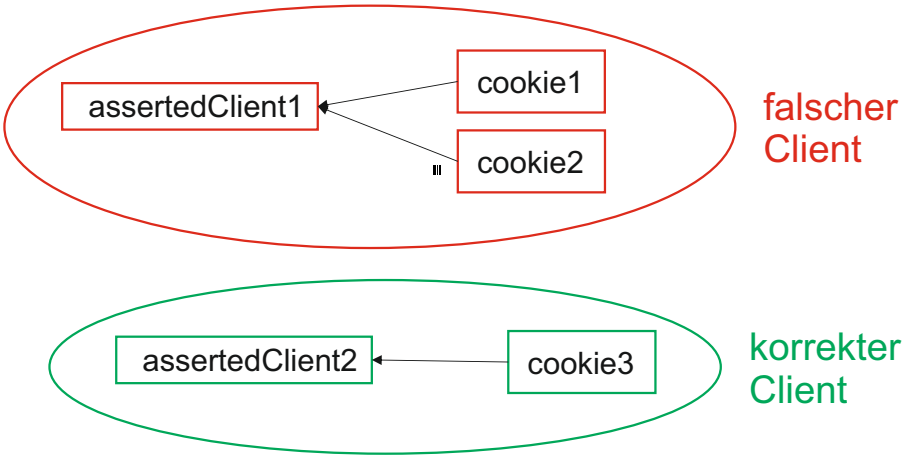

Abbildung 3.8: Client-Map

Der darauf aufbauende DOS wird folgendermaßen berechnet:

degreeOfSoftness (DOS)

$$degreeOfSoftness = wrongClientsClientMap/GesamtanzahlClients \tag{3.2}$$

Wenn der DOS sinkt, kann dies folgende Ursachen haben:

- Es ist eine geringere Löschung von Cookies erfolgt,
- die Wiedererkennung durch den Fingerprint-Service hat sich verschlechtert oder
- es ist eine Mischung von beidem.

Im Fall einer Steigung verhält es sich ebenfalls vice versa.

Korrelationsmetriken

Intention

Um eine Aussage trotz der Unzuverlässigkeit beider Verfahren treffen zu können, werden beide Verfahren zusammen betrachtet. Sollte nur ein AssertedClient einer Cookie-ID in der Cookie-Map zugeordnet sein und in der Client-Map nur dieser AssertedClient derselben Cookie-ID zugewiesen werden, dann kann mit höherer Wahrscheinlichkeit die Aussage getroffen werden, dass sowohl das Fingerprint-Verfahren als auch das cookie-basierte Verfahren korrekt gearbeitet haben und somit stellen beide Verfahren aufgrund der statistischen Masse aussagekräftige Metriken zur Verfügung. Sollte man aufgrund des DOS die ungefähre Tendenz der Löschrate der Cookies evaluiert haben, stellen diese Metriken nach der Korrektur eine bessere Aussage als der DOH zur Verfügung. Eine generelle Aussage hierfür kann aber nicht getroffen werden. Des Weiteren können auch Studien zum Löschverhalten und zur Verweigerung von Cookies einbezogen werden, um die Metriken der Korrelation besser einordnen und korrigieren zu können. Auch Umfragen unter den Benutzern des Angebots können darüber Aufschluss bieten.

correctCookiesFactor (CCF) / rightCookiesFactor (RCF)

$$correctCookiesFactor = correctCookies/GesamtanzahlCookies \tag{3.3}$$

Wenn der CCF steigt, kann dies folgende Ursachen haben:

- Es ist eine geringere Löschung von Cookies erfolgt,
- die Wiedererkennung durch den Fingerprint-Service hat sich verbessert oder
- es ist eine Mischung von beidem.

Im Fall einer Senkung verhält es sich vice versa.

correctVisitsFactor (CVF) / rightVisitsFactor (RVF)

$$correctVisitsFactor = correctVisits/GesamtanzahlVisits \tag{3.4}$$

Konklusion

Es ist sehr komplex, Aussagen über die Wiedererkennung treffen zu können. Ein großes Problem stellen die Mischverhältnisse dar. Durch logische Kombination von DOH, DOS und CCF ist dies aber in einem gewissen Rahmen möglich. Hierbei ist zu beachten, dass der CCF die Untergrenze und $1 - DOH$ die Obergrenze der client-basierten Wiedererkennungsrate (Clients, welche immer wieder korrekt erkannt wurden) darstellen. Beim Vergleich verschiedener Verfahren stellt der DOH die Aussage für die Vergleichbarkeit der Verfahren zur Verfügung. Dies ist darin begründet, dass der DOH zwar durch die Löschung der Cookies überbewertet ist und der korrigierte DOH, bzw. der CCF eine wesentlich bessere Aussage über die Wiedererkennungsrate treffen können, aber bei der Vergleichbarkeit möglichst wenige Fehlereinflüsse zählen. Im Endeffekt wird die Grundgesamtheit der Cookie-ID zu AssertedClient Zuordnungen, welche durch die Entfernung von AssertedClients schon besser ausfällt, um weitere insgesamt positiv ausfallende, aber der gleichen Verteilung unterliegenden Zuordnungen ergänzt. Positivere Wiedererkennungsraten werden beim DOH immens verstärkt. Dadurch geht aber nicht die Ordnung verloren. Sollte die Wiedererkennung aber deutlich unter 50 % liegen, tritt ein disparater Effekt ein und die oben getroffenen Aussagen verlieren ihre Gültigkeit.

3.3 Aggregation

Die Aggregation, wie in den Grundlagen beschrieben, wird nur auf Visits innerhalb eines Cookies angewendet. Dies ist darin begründet, dass damit verhindert werden soll, dass sich Clients, deren Visits immer die gleichen Merkmale beinhalten, auf die Evaluierung auswirken. Sinn und Zweck der Optimierungen liegen schließlich darin, sich verändernde Visits wieder dem richtigen Client zuzuordnen. Durch die Aggregation wird die Anzahl der Visits um 12.522 verringert.

3.4 Entwicklung des Fingerprint-Evaluierungstools (FET)

Das Fingerprint-Evaluierungstool ist ein anwendungspezifisches Tool, welches explizit auf die Optimierung des Fingerprint-Services ausgerichtet ist.

Funktionalität

Hauptaufgabe des FET ist die Durchführung des Fingerprintverfahrens mit verschiedenen Optimierungen sowie die Evaluierung dieser. Hierbei werden die eigentlichen Fingerprints ohne Bewertung (AssertedClient) durch das Fingerprint-Verfahren in einer

eigenen Column Family gesichert, anstatt diese über den Restful-Webservice einzubringen.

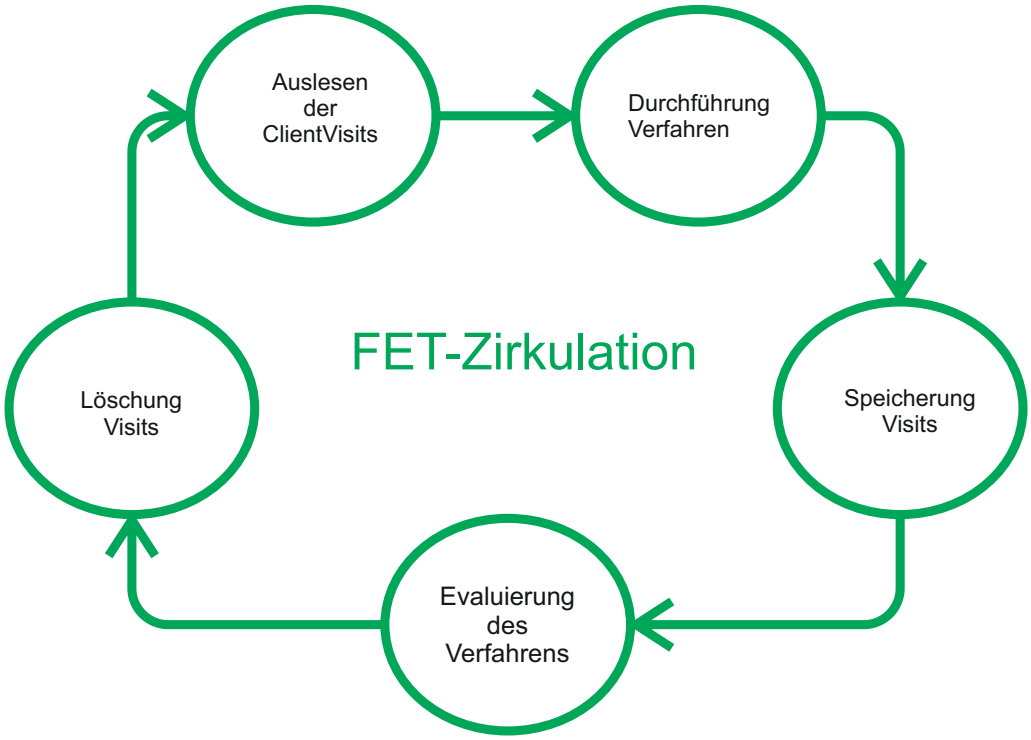

Abbildung 3.9: FET-Zirkulation

Des Weiteren bietet es Schnittstellen, mit denen auch auf andere Art und Weise fingerprint-spezifische Daten eingebracht und ausgegeben werden können. Momentan sind CSV, Cassandra-Import, bzw. Export und weitere IO-Funktionen implementiert. Folgende Funktionalitäten abseits der oben dargestellten Zirkulation werden momentan unterstützt:

- Filterung von ClientVisits nach verschiedenen Kriterien
- Datenaufbereitung für Gewichtung nach der Transinformation
- Setzen von Gewichtungen
- Evaluierung der Menge an Fingerprints mit gleichen Attributen
- Durchführung von Aggregation
- Löschung von ClientVisits
- Erstellung und Manipulation von Column Families
- Reporting in Form von CSV-Dateien
- Sortierung von ClientVisits
- Erstellung von sonstigen Statistiken (z. B. bezüglich der Attribute)

Systemvoraussetzungen

Minimal

- 8 GB Hauptspeicher
- 10 GB Festplattenspeicher
- JRE 1.7

Dabei ist die Verarbeitung von bis zu 600.000 ClientVisits garantiert. Diese Reglementierung ist dadurch begründet, dass alle ClientVisits in den Hauptspeicher geladen werden.

Optimal

- 16 GB Hauptspeicher
- 10 GB Festplattenspeicher
- JRE 1.7

Prozessmodell

Jeder Prozess besteht mindestens aus folgenden Komponenten:

Prozesskomponente	Beschreibung
Import	Import der für den Prozess notwendigen Daten
DataPreparation	Datenaufbereitung
Model	Durchführung der eigentlichen Aufgabe
Evaluation	Evaluierung
Export	Export von gewonnenen Daten

Tabelle 3.1: FET Prozessmodell

Maßgebende Implementierungsdetails

Speicherung der Fingerprints als ClientVisits

Da das Fingerprintverfahren in das FET übernommen wurde, müssen die Column Families zu Beginn des Verfahrens leer sein. Aus diesem Grund werden die Fingerprint-Daten zu sogenannten ClientVisits gemappt. Dies bedeutet, dass alle Einzelkomponenten eines Visits, welche aus der Cassandra Datenbank geholt werden, in einem Objekt zusammengefasst werden.

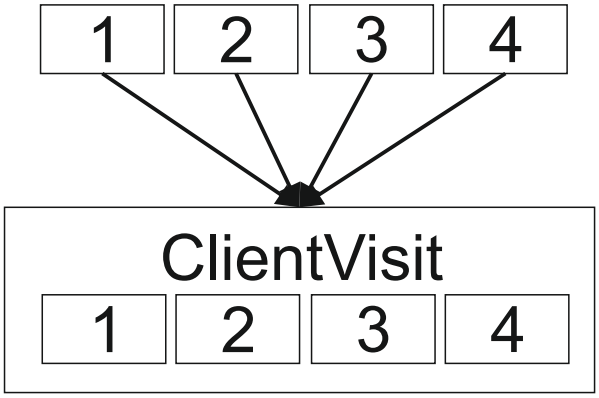

Abbildung 3.10: Mapping ClientVisit

Unterschied zum originalen Wiedererkennungsverfahren

Zu beachten ist, dass das ursprüngliche Wiedererkennungsverfahren asynchron durchgeführt wird. Das heißt, dass sowohl der Ähnlichkeitsvergleich als auch die Clientzuweisung mehrerer Visits gleichzeitig geschehen. Eine Auswirkung z. B. ist, dass bei jeder Durchführung des Verfahrens bei einem Visit unterschiedliche Visits (Clients) für die Klassifizierung zur Verfügung stehen. Aus diesem Grund erfordert die Evaluierung des Verfahrens die Umstellung auf eine synchrone Durchführung. Um für alle Optimierungen des Wiedererkennungsverfahrens die gleichen Bedingungen zu schaffen, werden die Visits zuerst anhand ihrer VisitTime sortiert und dann sequentiell in das Verfahren eingebracht.

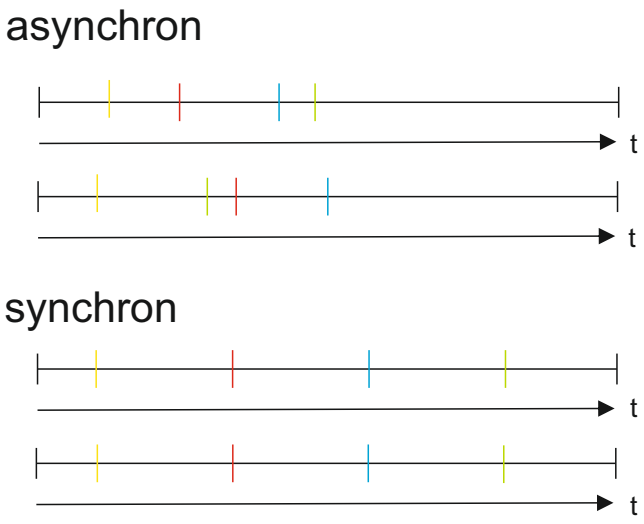

Abbildung 3.11: synchrones vs. asynchrones Verfahren der Wiedererkennung

4 Aktuelles Verfahren

Der Prozess des aktuellen Verfahrens ist in den Grundlagen unter dem Punkt 2.5. beschrieben. Um den Wiedererkennungsvorgang im Detail zu verstehen, wird auf die Art des binären Mappings eingegangen.

4.1 Funktionsweise aktuelles Identifizierungsverfahren

Binäres Mapping der Attribute

Vor der Durchführung des Ähnlichkeitsvergleichs wird zuerst die Menge der auf Zahlen gemappten Attributmerkmale ermittelt. Dabei handelt es sich um Attributmerkmale, denen natürliche Zahlen zugeordnet sind (z. B. Java $15.019 \Rightarrow 5$). Dies dient als Ausgangslage für das binäre Mapping. Es wird für jedes Attributmerkmal, das entweder in einem oder anderen Vektor vorhanden ist, geprüft, ob es im aktuell zu mappenden Vektor (Fingerprint) vorhanden ist. Wenn dies der Fall ist, wird dem binären Vektor (Ergebnisvektor des Mappings) eine Eins, ansonsten eine Null hinzugefügt.

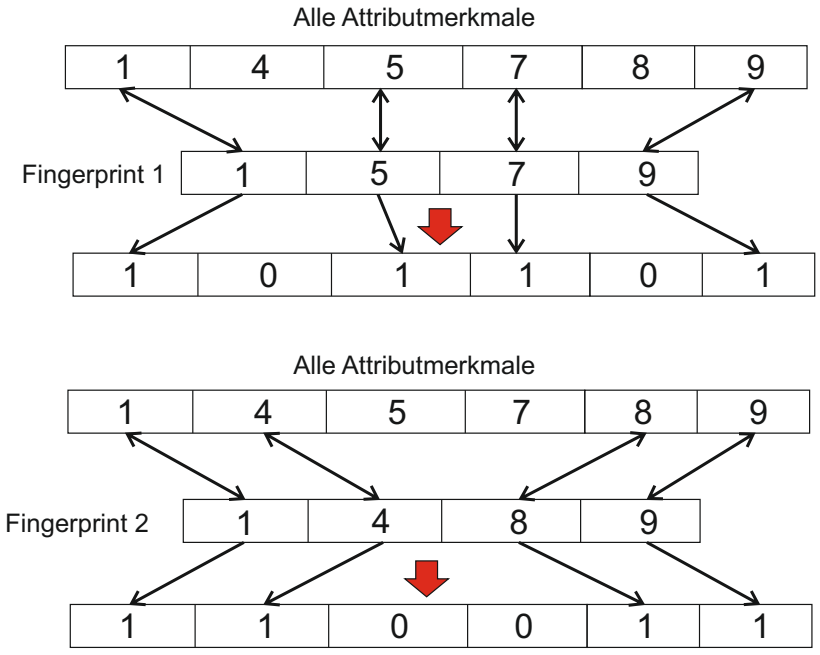

Abbildung 4.1: Binäres Mapping der Attribute

4.2 Klassifizierung eines neuen Visits

Ausgehend vom laufenden Betrieb befinden sich die Attribute der vorhandenen Visits im Hauptspeicher. Sollte ein neuer Visit hinzukommen, wird dieser mit jedem im System (nach binärem Mapping) verglichen. Sollte ein Ähnlichkeitsvergleich eine Übereinstimmung von 0,9999 oder höher ergeben, wird der Vorgang abgebrochen und der neue Visit dem Client des Visits, mit dem er zuletzt verglichen worden ist, zugewiesen. Ansonsten wird der Visit als Resultat des besten Ähnlichkeitsvergleichs gewählt. Wenn dessen Resultat höher als die vorgegebene Ähnlichkeitsschwelle ist, wird ebenso wie bei einer Ähnlichkeit von 0,9999 verfahren. Ansonsten wird für den Visit einer neuer Client erstellt.

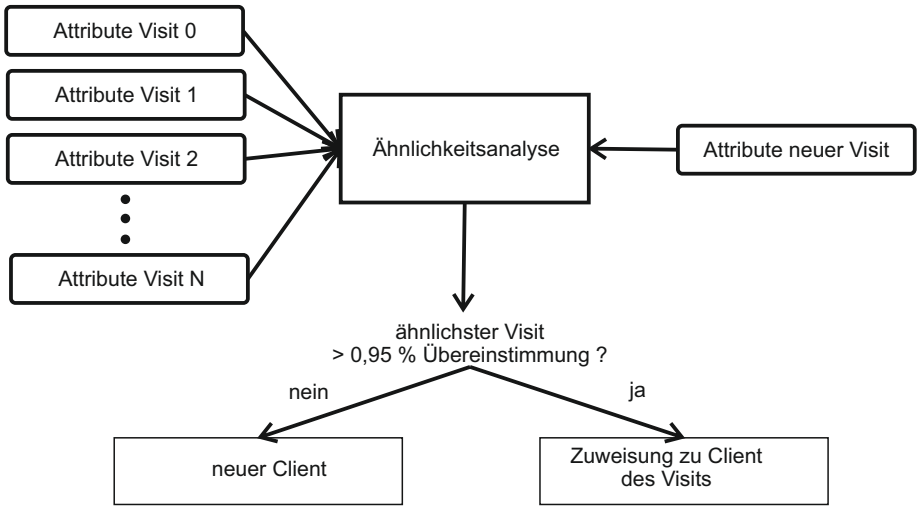

Abbildung 4.2: Klassifizierung eines neuen Visits

4.3 Evaluierung der Güte

Ergebnis abhängig von Mindestanzahl Visits

Mind. Visits	DOH	DOS	CCF	CVF
1	0,0819	0,0847	0,6640	0,4473
2	0,1517	0,1528	0,5784	0,4122
3	0,2026	0,2016	0,5250	0,3858
4	0,2416	0,2314	0,4949	0,3687
5	0,2757	0,2568	0,4688	0,3537
6	0,3058	0,2775	0,4460	0,3407
7	0,3325	0,2960	0,4298	0,3301
8	0,3567	0,3131	0,4147	0,3201
9	0,3796	0,3266	0,4012	0,3110
10	0,3994	0,3389	0,3873	0,3021
11	0,4161	0,3524	0,3757	0,2942
12	0,4324	0,3644	0,3633	0,2862

Tabelle 4.1: Von der Mindestlänge abhängiges Evaluierungsergebnis des aktuellen Verfahrens

Bei Betrachtung der Mindestanzahl von Visits fällt die sehr hohe Fehlerrate auf. Selbst wenn man im Schnitt von einem Visit pro Session ausgeht, was z. B. bei einer Neuregistrierung im Portal nicht der Fall ist, sind nach 12 und mehr Besuchen nach Betrachtung des DOH reichlich die Hälfte der Wiedererkennungsvorgänge des Fingerprintverfahrens korrekt. Nach den Korrelationsmetriken sind es sogar nur noch ungefähr ein Viertel.

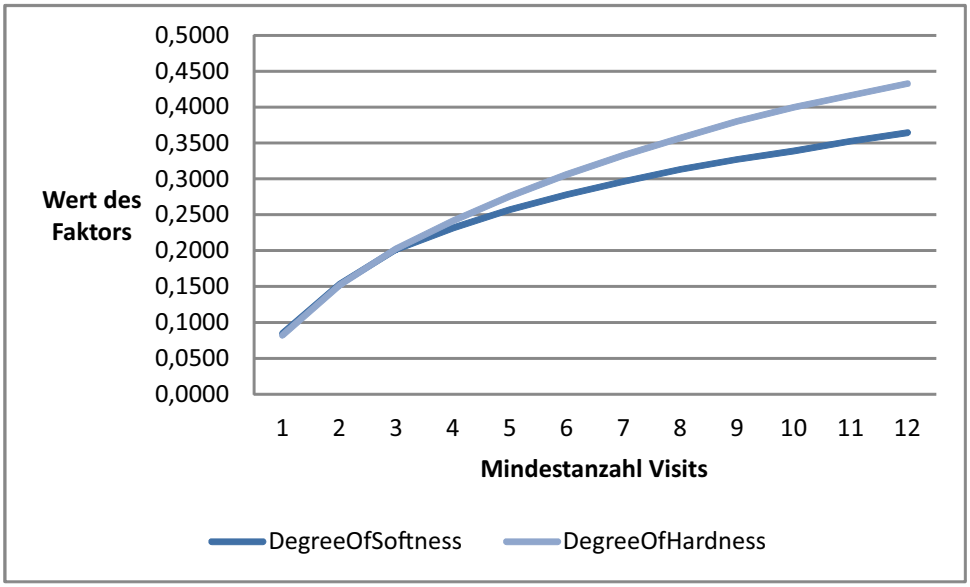

Abbildung 4.3: Aktuelles Verfahren: DegreeOfSoftness und DegreeOfHardness

Das Verhältnis der Mindestanzahl Visits und der Faktoren DOS und DOH verhält sich logarithmisch. Die Anzahl der Wiederkennungsfehler mit zunehmender Mindestlänge der Visits steigt zu Beginn (bis zur Mindestlänge 4) schnell an, erhöht sich aber bei zunehmender Mindestlänge nur noch minimal.

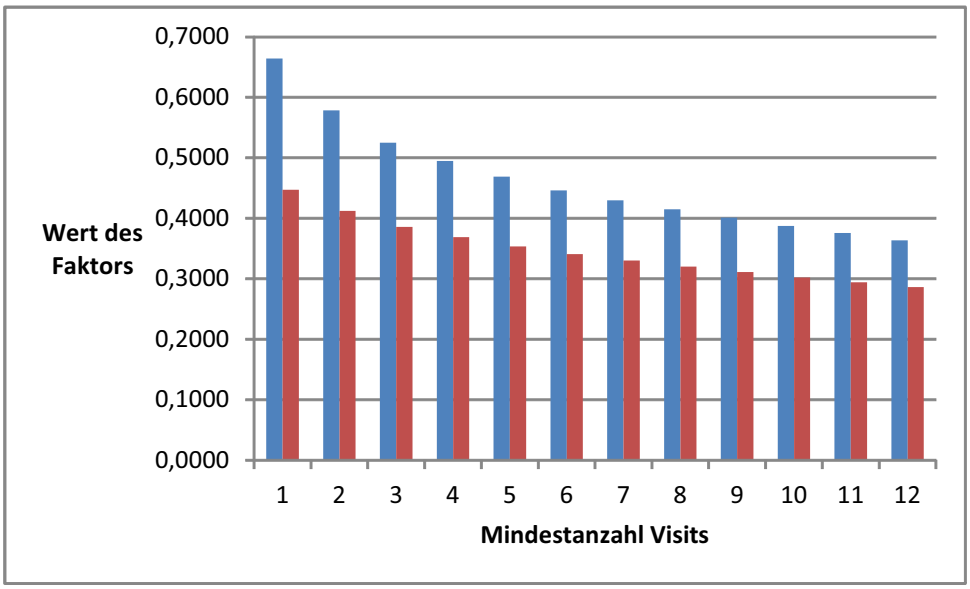

Abbildung 4.4: Aktuelles Verfahren: correctCookiesFactor und correctVisitsFactor

Wie erwartet, verhalten sich der CCF und der CVF umgekehrt zum Logarithmus von DOH und DOS. Aus dem Diagramm kann entnommen werden, dass die Güte des Wiedererkennungsverfahrens bei einer Mindestanzahl Visits anfangs sehr hoch ist, dann rapide fällt und im weiteren Verlauf abflacht. Dadurch kann impliziert werden, dass es für die Wiederkennung ab einer Länge von 12 so gut wie nicht mehr von Belang ist, wie oft ein Client nochmals wiederkehrt.

Ergebnis nach Aggregation

Das Evaluierungsverfahren wird nach der Filterung der durch die Aggregation [vgl. 3.3] ermittelten Visits durchgeführt.

DOH	DOS	CCF	CVF
0,1510	0,1529	0,5794	0,4207

Tabelle 4.2: Aktuelles Verfahren: Alle Faktoren

Über alle Attributlängen ergibt sich eine client-basierte Wiedererkennungsrate zwischen 42 % und 84,9 %. Aus dem Verhältnis zwischen CCF und CVF lässt sich erkennen, dass

die visit-basierte Wiedererkennung noch schlechter abschneidet. Dies bestätigt nochmals die Aussage, dass die Wiedererkennung sehr stark von der Anzahl der Besuche abhängt.

Ergebnis abhängig von der Attributlänge

Attributlänge	DOH	DOS	CCF	CVF
0-10	0,2285	0,2753	0,5092	0,3574
11-20	0,1447	0,3319	0,4595	0,3711
21-30	0,1079	0,1543	0,6649	0,5356
31-40	0,0755	0,1031	0,7225	0,6556
41-50	0,0000	0,3056	0,6250	0,7479
51-60	0,0425	0,2979	0,4205	0,3749
61-70	0,0138	0,2259	0,5160	0,4858

Tabelle 4.3: Aktuelles Verfahren: Alle Faktoren per Attributlänge

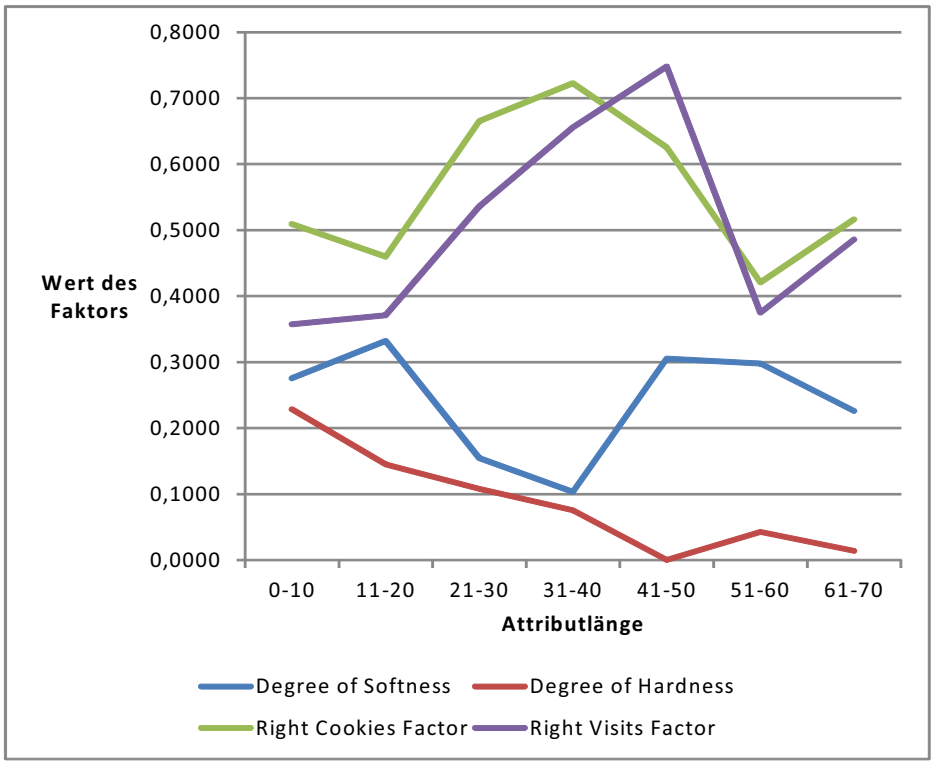

Abbildung 4.5: Aktuelles Verfahren: alle Faktoren

Konklusion

Bis zu einer Attributlänge von 35 sinken DOS und DOH synchron, während der CCF steigt. Dies impliziert entweder eine besser werdende Wiedererkennung oder eine geringere Löschung von Cookies. Aufgrund der Annahme, dass bei Clients mit kurzen Visits JavaScript deaktiviert wurde und diese somit sehr datenschutzaffin sind, wird daraus gefolgert, dass ein Großteil der Steigung durch eine geringere Löschung von Cookies verursacht wird. Bei Attributen der Länge 30 - 70 zeigt sich eine entgegengesetzte Entwicklung von DOH und DOS, während der CCF fällt. Dies deutet auf eine mit der Länge der Attribute schlechter werdende Wiedererkennung oder eine erhöhte Löschung von Cookies hin. Hieraus wird gefolgert, dass dies auf eine schlechter werdende Wiedererkennung zurückzuführen ist. Die Begründung dafür ist die Annahme, dass es bei Visits mit längeren Attributen keine großen Schwankungen in der Anzahl der Löschungen von Cookies gibt. Insgesamt betrachtet liefert das Cosinus Verfahren mit einem CCF von 0,5794 eine halbwegs gute client-basierte Wiedererkennung. Korrigiert man diese um die angenommene Schwankung der Attribute bis zur Länge von 35, dann kann man annehmen, dass mindestens 60 % aller Clients mit dem Cosinus-Verfahren korrekt wiedererkannt werden. Unterstützt werden diese Aussagen vom CVF.

5 Theoretische Ansätze zur Verbesserung des Verfahrens

5.1 Weitere Ähnlichkeits- und Distanzmaße

Da aktuell das Cosinus-Verfahren implementiert ist, fällt der erste Blick natürlich auf andere Maße zur Ähnlichkeits- oder Distanzbestimmung.

Ähnlichkeitsmaße

Die Wahl des optimalen Ähnlichkeitsmaßes hängt sehr stark von der Symmetrie der binären Variablen ab. Dabei gilt es als asymmetrisch, wenn nur ein Merkmal von Bedeutung ist. Ein symmetrisches Merkmal wäre z. B. das Geschlecht. Im Fall des Fingerprint-Verfahrens spielt es hauptsächlich eine Rolle, ob ein Attributmerkmal vorhanden ist [vgl. 17]. Für symmetrische Verfahren eignen sich am besten:

- Simple Matching
- Hamann Koeffizient
- Rogers

Da das Fingerprint-Verfahren asymmetrisch ist, werden folgende Ähnlichkeitsmaße theoretisch bevorzugt:

- Cosinus
- Dice
- Jaccard

Hierbei unterscheiden sich die verschiedenen Verfahren in der Gewichtung der Übereinstimmungen, beziehungsweise der Nicht-Übereinstimmungen. Beim Dice-Koeffizienten werden z. B. die Übereinstimmungen doppelt gewichtet, während der Jaccard-Koeffizient beide Fälle gleich gewichtet.

Über die restlichen Verfahren kann keine Aussage aufgrund von Erfahrungen oder Literatur getroffen werden. Aus diesem Grund werden diese in der Praxis getestet.

Distanzmaße

Distanzmaße sind im Allgemeinen für metrische Variablen geeignet, was hier auf Grund der binären Darstellung nicht der Fall ist. Inwieweit sich diese auch für binäre Variablen für den Fingerprint-Service eignen, wird im nächsten Kapitel betrachtet.

5.2 Einsatz von Proximitätsmaßen je nach Attributlänge eines Visits

Falls verschiedene Proximitätsmaße je nach Attributlänge verschiedene Grade der Wiedererkennungsgüte aufweisen, könnte die Qualität insgesamt verbessert werden, indem je nach Anzahl der Attribute eines Visits verschiedene Maße angewendet werden.

5.3 Wichtungen

Gewichtung der Attributausprägungen aufgrund von Transinformation

Bei dieser Art der Gewichtung wird versucht, die Wertigkeit der Attributausprägung für die Trennung der Attribute festzustellen. Hierbei wird die Ungleichmäßigkeit der Verteilung der Nullen (nicht vorhanden) und Einsen (vorhanden) auf die einzelnen Cookie-IDs betrachtet. Je häufiger ein Attributmerkmal (zum Beispiel 1) in einer Cookie-ID und je weniger es in den anderen vorhanden ist, desto höher fällt der Wert eines Attributs aus [vgl. 11]. Wenn man beispielsweise die Verteilung der Betriebssysteme in Augenschein nimmt, erkennt man, welche Vorteile dies bietet. Sollte zum Beispiel jeder Visit die Attributausprägung Windows 7 aufweisen, ist dieses als Unterscheidungsmerkmal wenig geeignet. Sollte es so gut wie überhaupt nicht auftauchen, ist es ebenso schlecht als Merkmal zur Unterscheidung geeignet. Gibt es jedoch nur eine Cookie-ID mit dem Betriebssystem Plan 9, erzielt dieses Attributmerkmal einen sehr hohen Wert, weil damit der Client eindeutig identifiziert werden kann.

Zum besseren Verständnis folgt eine Beispielrechnung:

Klassifizierungsmerkmale: c1, c2, c3

CookieID	Flash 11.3.300.268	Adobe Reader 10.010	Java 16.024	Java 15.019
c1	0	1	1	0
c2	1	1	1	0
c1	1	1	0	0
c3	1	1	0	0
c1	0	1	1	0
c2	1	1	1	0
c3	1	1	1	0

Tabelle 5.1: Daten für Beispiel der Gewichtung nach der Transinformation

Attributmerkmal	Verteilung 0	Verteilung 1
Flash 11.3.300.268	c1:2 c2:0 c3:0	c1:1 c2:2 c3:2
Adobe Reader 10.010	c1:0 c2:0 c3:0	c1:3 c2:2 c3:2
Java 16.024	c1:2 c2:0 c3:1	c1:1 c2:2 c3:1
Java 15.019	c1:3 c2:2 c3:2	c1:0 c2:0 c3:0
Gesamt	c1:6 c2:2 c3:3	c1:6 c2:6 c3:5

Tabelle 5.2: Verteilung der Attributmerkmale

$$H(ges) = -\frac{3}{7} * log_2(\frac{3}{7}) - \frac{2}{7} * log_2(\frac{2}{7}) - \frac{2}{7} * log_2(\frac{2}{7})) = 1,5567 \tag{5.1}$$

$$H(flash0) = \frac{2}{7} * (-\frac{2}{2} * log_2(\frac{2}{2}) - \frac{0}{2} * log_2(\frac{0}{2}) - \frac{0}{2} * log_2(\frac{0}{2}))) = 0 \tag{5.2}$$

$$H(flash1) = \frac{5}{7} * (-\frac{1}{5} * log_2(\frac{1}{5}) - \frac{2}{5} * log_2(\frac{2}{5}) - \frac{2}{5} * log_2(\frac{2}{5}))) = 1,0871 \tag{5.3}$$

$$H(ges|flash) = 0 + 1,0871 = 1,0871 \tag{5.4}$$

$$g(flash) = H(ges) - H(ges|flash) = 1,5567 - 1,0871 = 0,470 \tag{5.5}$$

Attributmerkmal	information gain
Flash	0,470
Adobe Reader	0
Java 16.024	0,184
Java 15.019	0

Tabelle 5.3: Transinformationswerte der Beispielmerkmale

Flash 11.3.300.268 > Java 16.024 > Java 15.019 == Adobe Reader 10.010

Gewichtung der Attributausprägungen aufgrund der Korrelation

Die Korrelation zwischen den Merkmalen kann, wie im Abschnitt 2.3. beschrieben, mit Hilfe des Chi-Quadrat-Unabhängigkeitstest festgestellt werden. Hierfür müssen ein anderes Beispiel, bzw. andere Häufigkeiten gewählt werden, da dieser Test nur bei erwarteten Häufigkeiten von größer als fünf durchgeführt werden kann. Da der Test nur theoretisch durchgeführt wird, werden bei dem folgenden Beispiel zwei Merkmale miteinander verglichen und keine Häufigkeits- und Gewichtungsmatrizen gebildet. Es werden die Merkmalsvariablen A für den Adobe Reader und F für Flash festgelegt:

	AdobeR - 0	AdobeR - 1	Summe
Flash 11 - 0	123	13	136
Flash 11 - 1	12	55	67
Summe	135	68	203

Tabelle 5.4: Häufigkeiten für den Chi-Quadrat-Unabhängigkeitstests

Bei folgenden Berechnungen wird von der Nullhypothese, dass beide statistisch unabhängig sind, ausgegangen. Aus den Häufigkeiten lassen sich die erwarteten Häufigkeiten berechnen. Als Beispiel für $h_{af} = h_{12}$ ergibt das:

$$h_{12} = \frac{h_a * h_f}{n} = \frac{h_1 * h_2}{n} = \frac{136 * 68}{203} = 45,56 \tag{5.6}$$

Dadurch erhält man folgende Kreuztabelle:

	AdobeR - 0	AdobeR 1	Summe
Flash 11 - 0	90,44	45,56	136
Flash 11 - 1	44,56	22,44	67
Summe	135	68	203

Tabelle 5.5: Kreuztabelle des Chi-Quadrat-Unabhängigkeitstests

Man kann nun die Korrelation zwischen den zwei Merkmalen Adobe und Flash folgendermaßen berechnen:

$$\chi^2 = \frac{123 - 90,44}{90,44} + \frac{13 - 45,56}{45,56} + \frac{12 - 44,56}{44,56} + \frac{55 - 22,44}{22,44} = 106.0271 \tag{5.7}$$

Zur Beurteilung und damit zum Vorgehen gegen die Nullhypothese wird ein Signifikanzniveau von 0,05 verwendet. Der kritische Wert der Prüfgröße ist 3,8413.

$$\chi^2(0,95,1) = 3,8413 > 106.0271 \tag{5.8}$$

Damit ist die Nullhypothese widerlegt und es liegt eine statistische Abhängigkeit vor. Aufgrund dessen kann man zum Beispiel die Gewichtung beider Merkmale senken (z. B. um 0,1). Dies kann aber auch mit besseren Ergebnissen, abhängig von der Prüfgröße gestaltet werden.

Attribut und Attributausprägungskorrelation

Verknüpfungen von Gewichtungen der Attributausprägung mit denen der Attribute.

5.4 Architekturänderungen

Erhebung zusätzlicher Attribute

Durch die Erhebung zusätzlicher Attribute mit sehr starkem Trennungsgehalt und/oder niedriger Änderungsrate kann die Auflösung weiter verbessert werden. Sehr viel versprechende Attribute hinsichtlich dieser Vorgabe sind:

- Ortsangaben
- IP (hohe Änderungsrate)
- nutzerspezifische Daten (z. B. Kundennummer)

Entfernung von Attributen

Die Attribute, deren Ausprägungen sich als schlechte Trennungsmerkmale erweisen oder eine hohe Korrelation aufweisen, können entfernt werden, was wiederum einer Gewichtung von 0 gleichzusetzen ist, aber aus Geschwindigkeitsgründen mehr Sinn ergeben würde.

5.5 Verschiebung der Ähnlichkeitsschwelle

Wie in Sektion 4.2. beschrieben, muss ein Visit beim Vergleich eine Ähnlichkeitsschwelle überwinden, um einem schon vorhandenen Client zugeordnet zu werden. Bei den momentanen Untersuchungen wird ein Schwellenwert von 0,95 verwendet. Diese Schwelle kann nach unten oder oben korrigiert werden. Die Auswirkungen einer Korrektur nach unten sind, dass dadurch zwar großzügigere Änderungen der Attributausprägungen toleriert werden, aber dies höchstwahrscheinlich die Trennung zwischen den Clients verwässert. Bei einer Korrektur nach oben verhält es sich vice versa. Die Untersuchungen hinsichtlich der optimalen Schwelle sind nicht Teil der Arbeit und werden nur theoretisch betrachtet.

5.6 Umstellung auf das metrische Skalenniveau

Da eine Transformation von nominalen zu metrischen Variablen nicht ohne Weiteres möglich ist, da diese normalerweise keine Ordnung beinhalten, müssen die Ausprägungen entsprechend codiert werden. Dies dürfte beispielsweise bei Plugins mit Versionierung oder geometrischen Daten kein Problem darstellen, wenn diese auf das gleiche Intervall normiert werden. Folgende Illustration zeigt eine einfache, wenn auch nicht optimale Möglichkeit, eine nominale, in dem Fall die Versionsnummer des Attributs Adobe Flash, zu einer metrischen Variable zu transformieren.

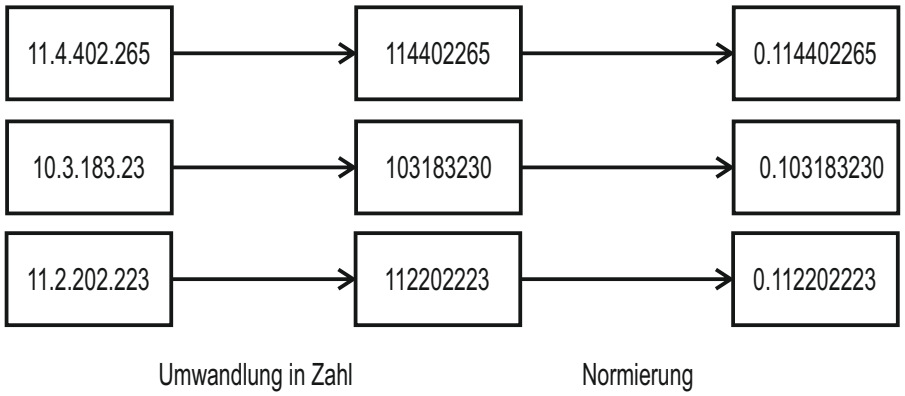

Abbildung 5.1: Metrisierungsmöglichkeit einer nominalen Variable

Sollte einem Attribut überhaupt keine Ordnung injiziert werden können, gibt es Möglichkeiten, diese durch Dummykodierung, Effektkodierung oder Kontrastkodierung künstlich herzustellen [vgl. 13].

6 Umsetzung und Auswertung der theoretischen Ansätze

6.1 Alternative Ähnlichkeits- und Distanzverfahren

Implementierung

Zum Austausch muss die Klasse VectorCosScpr (A3S-CF-Engine) der aktuellen Cosinus-Implementierung gegen eine andere Implementierung ausgetauscht werden. Hierfür wurde die Klasse VectorCosScpr weiterentwickelt und weitere Ähnlichkeits- und Distanzalgorithmen hinzugefügt. Der Quellcode von ConcurrentVectorProxScpr wird aufgrund der Übersicht und der fehlenden Notwendigkeit nicht in dieser Arbeit abgebildet.

Dies erreicht man durch folgende Codeänderungen in der Klasse VisitsRCR (A3S-Fraud).

In Zeile 46 muss die Deklaration des Objekts scpr

```
private ConcurrentVectorCosScpr scpr;
```

gegen

```
private ConcurrentVectorProxScpr scpr;
```

getauscht werden. Des Weiteren muss in Zeile 280 die Zuweisung der Instanz

```
scpr = new ConcurrentVectorCosScpr(0.0);
```

gegen

```
scpr = new ConcurrentVectorProxScpr(0.0);
```

ausgetauscht werden.

Ergebnisse

Dice-Koeffizient

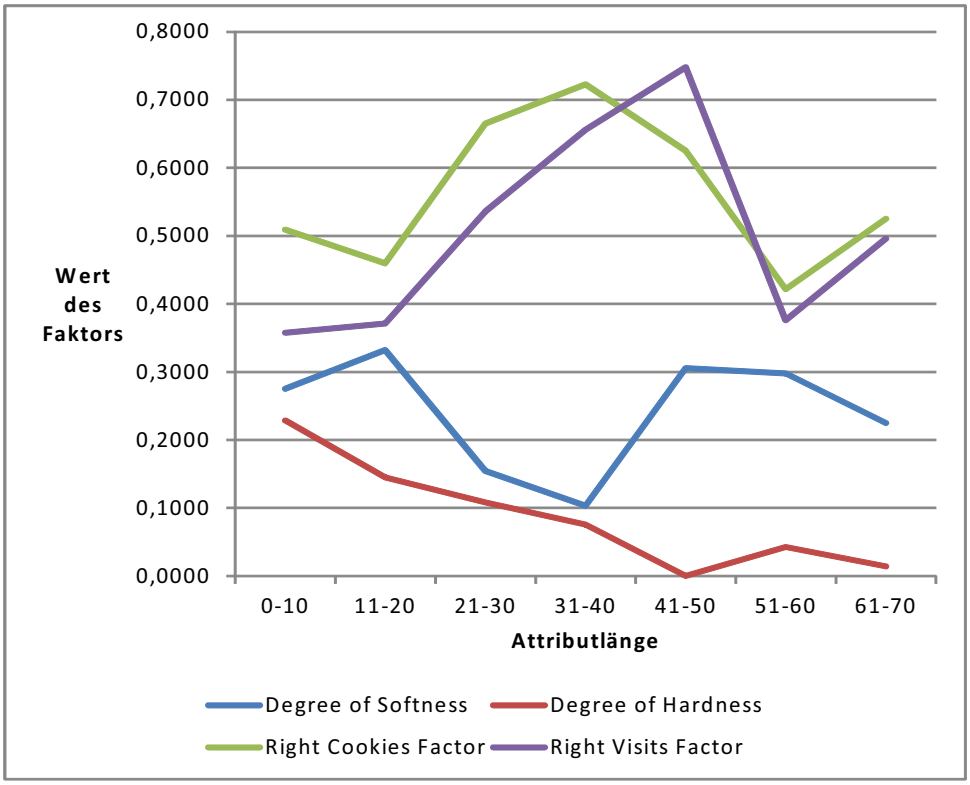

Abbildung 6.1: Dice: Alle Faktoren per Attributlänge

Attributlänge	DOH	DOS	CCF	CVF
0-10	0,2285	0,2753	0,5992	0,3574
11-20	0,1447	0,3319	0,4595	0,3711
21-30	0,1079	0,1543	0,6650	0,5357
31-40	0,0755	0,1030	0,7226	0,6557
41-50	0,0000	0,3056	0,625	0,7479
51-60	0,0425	0,2977	0,4217	0,3756
61-70	0,0141	0,2249	0,5249	0,4958

Tabelle 6.1: Dice: Alle Faktoren per Attributlänge

Das Verfahren des Dice-Koeffizienten verhält sich bis zu einer Attributlänge von 50 fast exakt wie das Cosinus Verfahren. Danach gibt es einige kleinere Fluktuationen zwischen beiden Verfahren. Der DOH liegt knapp über dem des Cosinus-Verfahren, es schneidet also minimal schlechter ab.

Jaccard-Koeffizient

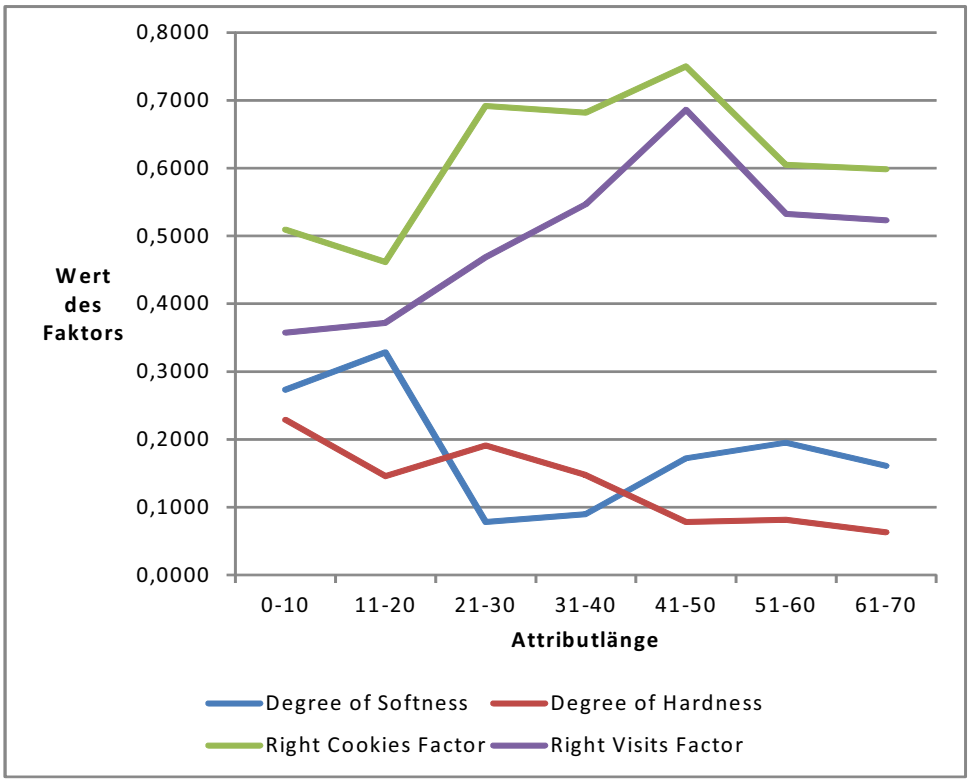

Abbildung 6.2: Jaccard: Alle Faktoren

Attributlänge	DOH	DOS	CCF	CVF
0-10	0,2285	0,2732	0,5092	0,3574
11-20	0,1455	0,3282	0,4616	0,3719
21-30	0,1908	0,0782	0,6917	0,4686
31-40	0,1474	0,0900	0,6817	0,5464
41-50	0,0781	0,1719	0,750	0,6860
51-60	0,0815	0,1950	0,6047	0,5324
61-70	0,0630	0,1607	0,5981	0,5232

Tabelle 6.2: Jaccard: Alle Faktoren per Attributlänge

Besondere Aufmerksamkeit ziehen die Attributspannen von 51-70 auf sich. Hier verfügt Jaccard über die Faktoren CCF und DOS, welche auf sehr hohem Niveau stagnieren. Dies deutet auf eine Überbewertung des DOH Jaccard-Verfahren ab einer Attributlänge von 51 und somit eine schlechtere Wiedererkennung hin. Auch bei diesem Verfahren muss bei der Feststellung der Wiedererkennungsrate (Spanne) bei der beim Cosinus-Verfahren geltenden Hypothese gefolgt werden und der CCF nach oben, bzw. der DOH nach unten korrigiert werden.

Overlap-Koeffizient

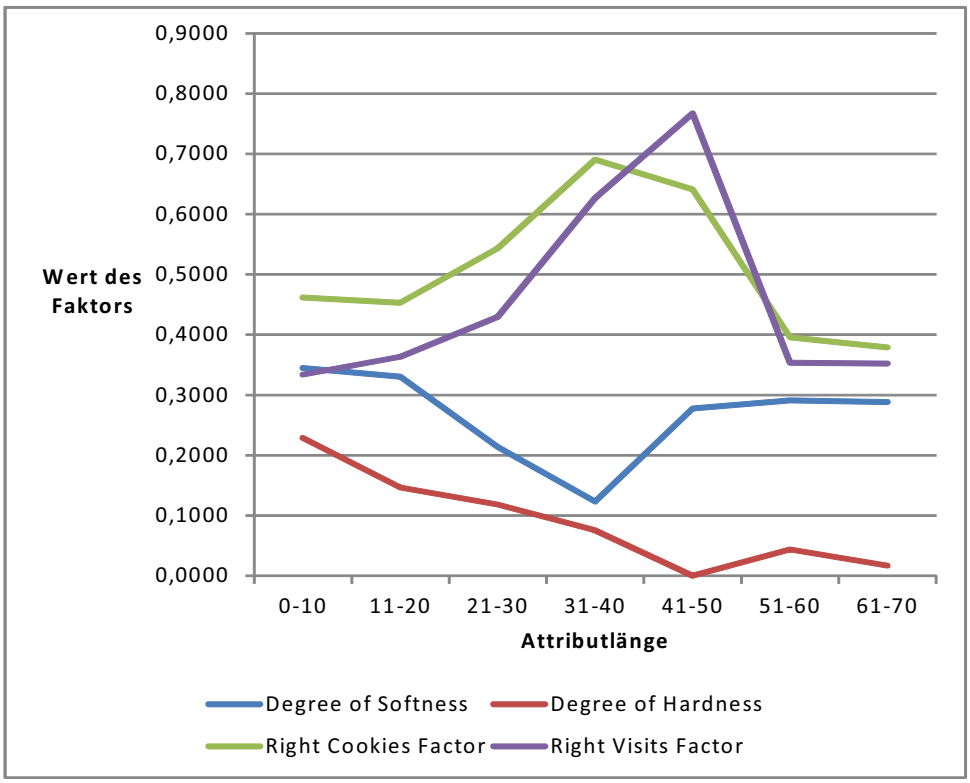

Abbildung 6.3: Overlap: Alle Faktoren

Attributlänge	DOH	DOS	CC	CV
0-10	0,2285	0,3446	0,4616	0,3337
11-20	0,1461	0,3302	0,4530	0,3633
21-30	0,1182	0,2138	0,5432	0,4293
31-40	0,0753	0,1233	0,6901	0,6265
41-50	0,0000	0,2778	0,6406	0,7670
51-60	0,0435	0,2911	0,3957	0,3536
61-70	0,0166	0,2882	0,3791	0,3522

Tabelle 6.3: Overlap: Alle Faktoren per Attributlänge

Die Struktur der Graphen des Overlap-Koeffizienten verfügen über den gleichen Verlauf wie das Cosinus-Verfahren, allerdings durchgehend auf niedrigerem Niveau.

Euklid-Koeffizient

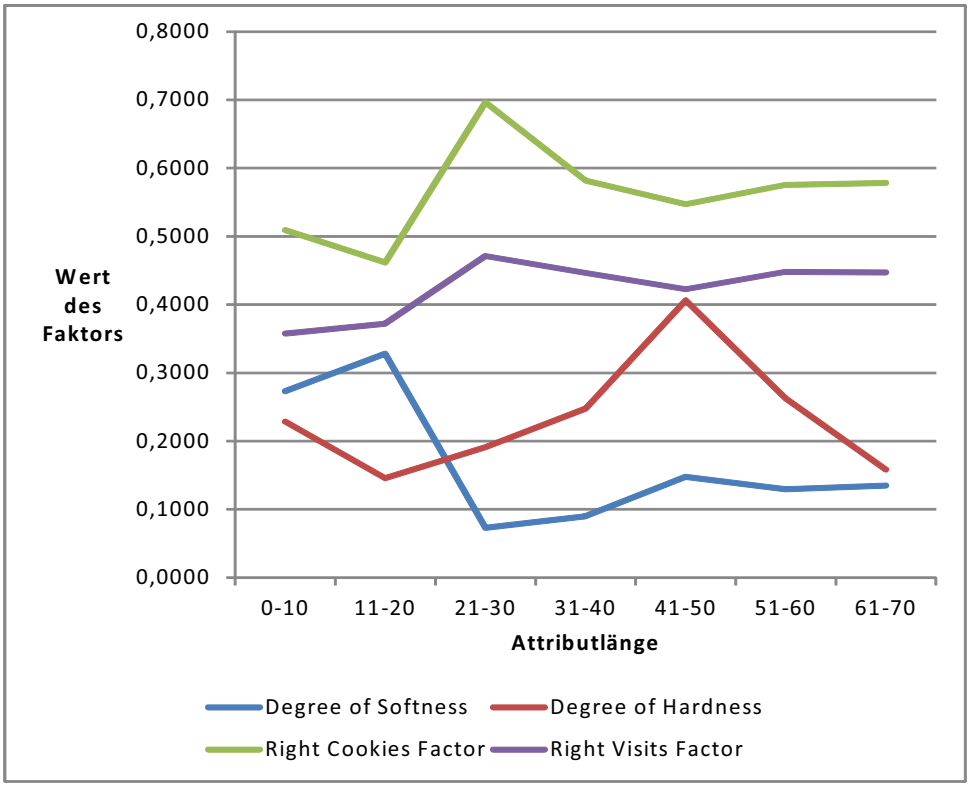

Abbildung 6.4: Euklid: Alle Faktoren

Wie angenommen lässt sich die Unbrauchbarkeit des Euklid-Distanzmaßes feststellen. Ein DOH von bis zu 0,4, d. h. eine Wiedererkennungsfehlerrate von 40 %, bei der die Cookielöschungsrate noch nicht enthalten ist, beweisen dies.

Attributlänge	DOH	DOS	CCF	CVF
0-10	0,2285	0,2732	0,5092	0,3574
11-20	0,1455	0,3278	0,4616	0,3719
21-30	0,1910	0,0731	0,6962	0,4712
31-40	0,2476	0,0897	0,5817	0,4461
41-50	0,4065	0,1478	0,5469	0,4225
51-60	0,2629	0,1296	0,5754	0,4478
61-70	0,5040	0,3377	0,5784	0,4473

Tabelle 6.4: Euklid: Alle Faktoren per Attributlänge

Alle Faktoren der gesamten Attributlänge in der Übersicht:

Proximitätsmaß	DOH	DOS	CCF	CVF
Dice	0,1511	0,1525	33708	183978
Jaccard	0,2610	0,1013	35496	169794
Euklid	0.2392	0.0801	35835	166627
Manhattan	0.2392	0.0801	35835	166627
Pearson	0.2392	0.0801	35835	166627
Overlap	0.1437	0.2229	24827	136411

Tabelle 6.5: Gesamt: Alle Faktoren per Attributlänge

Degrees of Hardness

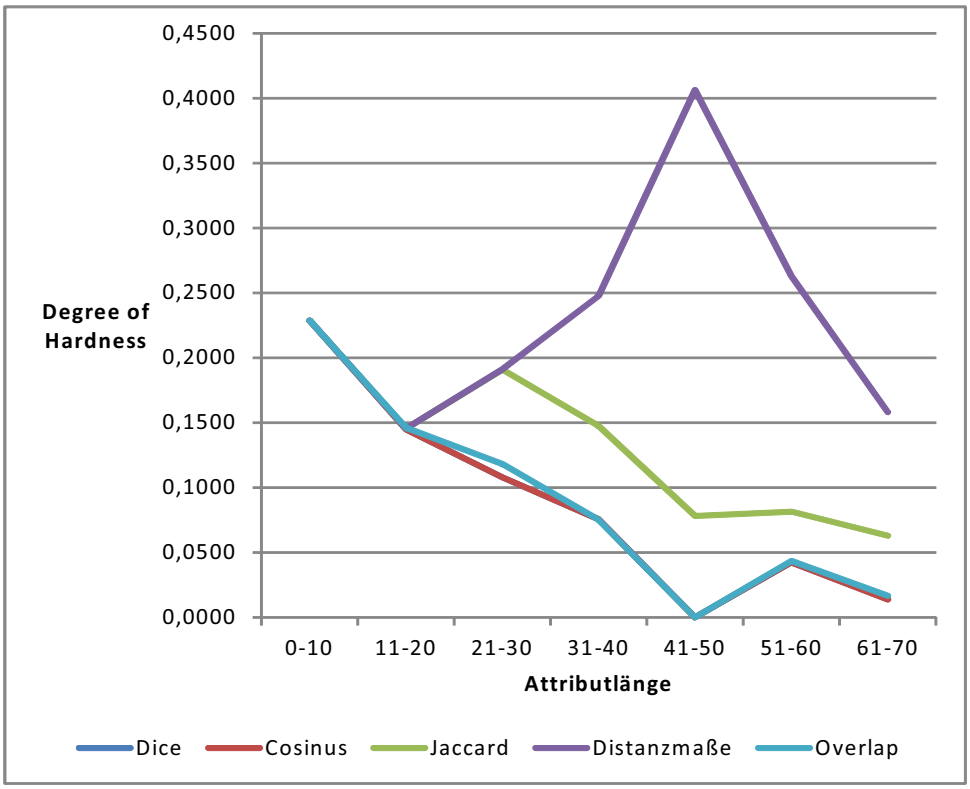

Abbildung 6.5: Alle evaluierten Verfahren: Degree of Hardness

Da bei dem Vergleich der Verfahren nur der DOH berücksichtigt wird, schneiden Dice, Cosinus und Overlap mit der gleichen Tendenz gut ab, wobei Cosinus den geringsten DOH hat. Das Jaccard-Verfahren liegt knapp darüber und folgt dem Trend. Die Distanzmaße in Form von Euklid und Manhattan sowie der Pearson-Koeffizient schneiden schlecht ab. Dies bestätigt die oben getätigte Aussage, dass Distanzmaße für binär nominale Variablen nicht geeignet sind, auch in diesem Fall. Das Cosinus-Ähnlichkeitsmaß stellt also das beste der getesteten Proximitätsmaße dar.

6.2 Gewichtung der Attribute

Praktische Umsetzung

Ermittlung information gain

Um die in Abschnitt 5.3. theoretisch beschriebene Gewichtung nach der Transinformation durchführen zu können, müssen zuerst die Gewichtungen aufgrund des Information Gain bestimmt werden. Dafür müssen die Daten entsprechend aufbereitet werden. Nach der Aggregation werden die ClientVisits der Tabelle ClientVisitStorage eingelesen und aus diesen die Mapped-IDs der Attributausprägungen und die zugehörige Cookie-ID extrahiert. Die Mapped-IDs werden daraufhin in einen Vektor aus Nullen (Attributausprägung nicht vorhanden) und Einsen (Attributausprägung vorhanden) umgewandelt.

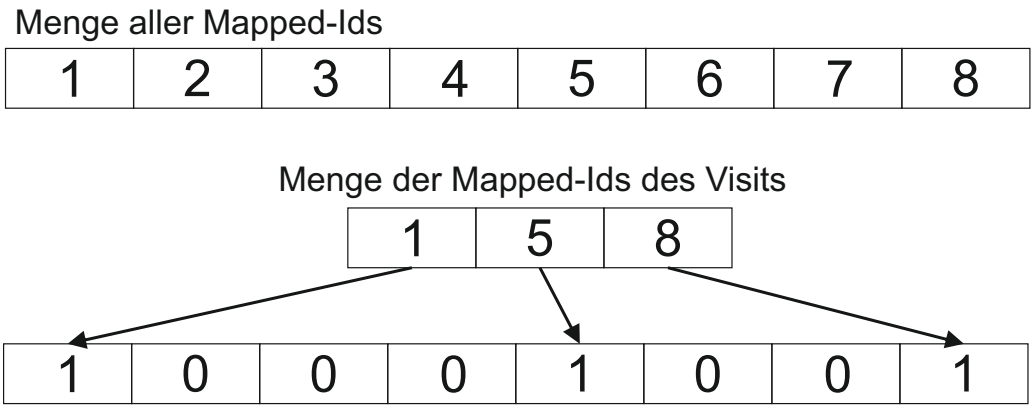

Abbildung 6.6: Datenaufbereitung für die Ermittlung der Transinformation (Information Gain)

Anschließend kann mit dem Tool Rapidminer [vgl. 2] folgender Prozess kreiert und damit der Gewichtungsvektor berechnet werden.

Abbildung 6.7: Prozess Rapidminer für die Ermittlung der Transinformation

Implementierung der Wichtungen

Attribute

Um die Attribute zu wichten, wird jedem Key der Column Family eine zusätzliche Spalte Weight hinzugefügt.

column	value
weight	value z. B. 0,83

Attributausprägungen

Um die Attributausprägungen zu wichten, wird eine neue Column Family erstellt. Diese als IdToWeight bezeichnete Column Family weist folgende Struktur auf:

key	column	value
mappedID	weight	value z. B. 0,83

Neben der Implementierung der Gewichtserweiterung in den Column Families muss weiterhin auch der Sourcecode angepasst werden, um die folgende Erweiterung der Distanz- und Ähnlichkeitsverfahren (z. B. cosinus) umzusetzen.

$$sim(x,y) = \frac{\sum_{i=1}^{n}(w_i^2 * x_i * y_i)}{\sqrt{\sum_{i=1}^{n}(w_i * x_i)^2} * \sqrt{\sum_{i=1}^{n}(w_i * y_i)^2}} \tag{6.1}$$

Zur Umsetzung der Gewichtung müssen in der Klasse VisitAnalyzer beide verwendeten

```
prefList.add(new Preference(id,aId,1));
```

gegen

```
prefList.add(new Preference(id,aId,mapper.
    getWeightForAttribute(aId)));
```

getauscht werden. Dies sorgt dafür, dass der Standardwert (1.0) gegen den Wert getauscht wird, welcher der Gewichtung für die entsprechende Attributausprägung entspricht. Das Setzen der Gewichte soll in dieser Arbeit nicht näher beschrieben werden, obwohl dies mit dem FET umgesetzt wurde.

Ergebnisse

Auf Grund der Größe der Fingerprintdaten und der damit verbundenen Verarbeitungszeit wurde für die Evaluierung der Gewichtungen eine Stichprobe der Größe 50.000 (vor Aggregation) verwendet. Die Evaluierung dieser hat für das Cosinus-Verfahren mit und ohne die Gewichtungen folgende Ergebnisse erbracht:

Proximitätsmaß	DOH	DOS	CCF	CVF
Cosinus mit Gewichtung	0,2767	0,3359	0,4557	0,3710
Cosinus ohne Gewichtung	0,2412	0,2149	0,6005	0,5030

Tabelle 6.6: Gesamt: Ergebnisse der Evaluierung des Cosinus-Verfahrens mit und ohne Transinformationsgewichtung

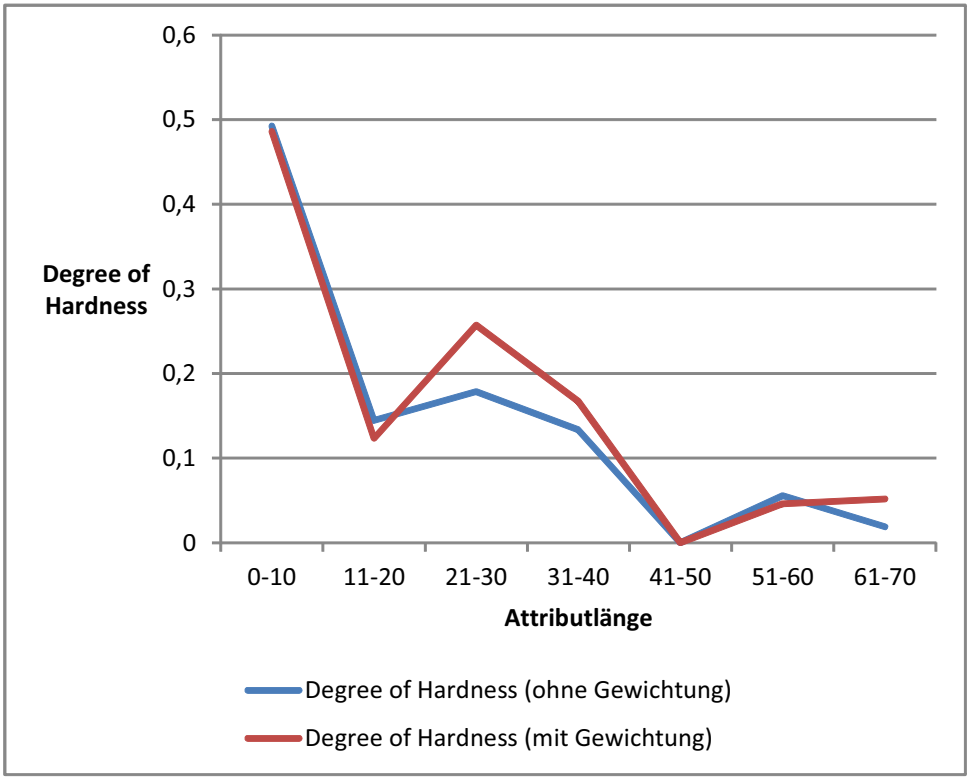

Abbildung 6.8: Ergebnisse der Evaluierung des Cosinus-Verfahrens mit und ohne Transinformationsgewichtung

Dies zeigt, dass die Gewichtung einer Attributausprägung aufgrund der Transinformation beim Cosinus-Verfahren eine schlechtere Wiedererkennung zur Folge hat, als wenn keine Gewichtung durchgeführt wird. Zur Verifizierung kann die Grundgesamtheit herangezogen werden. Daraus können aber noch keine Rückschlüsse auf die anderen Verfahren gezogen werden, obwohl eine Tendenz vermutet werden kann.

7 Zusammenfassung und Ausblick

In diesem Kapitel werden die Ergebnisse der Arbeit zusammengefasst, ein Ausblick auf künftige Untersuchungen zur Optimierung gegeben sowie ein Fazit bezüglich dieser Arbeit gezogen.

7.1 Zusammenfassung

Die allgemeinen Untersuchungen haben ergeben, dass die Wiederkehrrate der Nutzer des Webportals sehr hoch und damit für Untersuchungen optimal ist. Des Weiteren konnten die Häufigkeiten der Attributlängen ermittelt werden. Es wurden sowohl Metriken zur Einschätzung der Wiedererkennungsrate als auch zum Vergleich mit anderen Verfahren entwickelt. Es wurden auch theoretische Überlegungen getätigt, wie Optimierungen gestaltet werden könnten. Praktisch umgesetzt wurden sowohl die Ähnlichkeitsanalyse mit sieben verschiedenen Proximitätsmaßen als auch die Gewichtung nach der Transinformation.

7.2 Ausblick

Es könnten vier Richtungen weiterverfolgt werden.

Die erste wäre, weiterhin zu versuchen, das Verfahren auf Basis von binären Merkmalen weiterzuentwickeln. Der nächste Schritt dahingehend wäre, statt einer Stichprobe die Grundgesamtheit zu verwenden und die Transinformationsgewichtung für anderen Verfahren zu nutzen. Des Weiteren könnte eine Gewichtung auf Grund der Korrelation erfolgen.

Die zweite Richtung würde die Umstellung auf metrisches Skalenniveau bedeuten und damit fast zwangsläufig eine Nutzung von Distanzmaßen nach sich ziehen. Auch hier könnte versucht werden, mit Korrelation, Trennungsgehalt oder völlig anderen Methoden das Verfahren weiter zu optimieren.

Die dritte Richtung würde einen Ausbau der Architektur bedeuten, z. B. durch zusätzliche Merkmale wie den Aufenthaltsort.

Die letzte der Richtungen, welche verfolgt werden könnte, wäre die Kombination des Fingerprint-Services mit einer Cookie-basierten Wiedererkennung.

7.3 Fazit

Ziel der vorliegenden Arbeit war es, Metriken zur Feststellung der Wiedererkennungsrate zu entwickeln, verschiedene Möglichkeiten der Optimierung zu evaluieren sowie diese umzusetzen. Diese Ziele wurden erreicht, auch wenn das optionale Ziel, eine Optimierung zu finden, nicht erfüllt wurde. Auch wenn dies nicht der Fall ist, so wurden dennoch Voraussetzungen geschaffen, um die Optimierungen festzustellen und auch in gewissem Umfang das Verfahren mit anderen Verfahren (z. B. einem Wiedererkennungsverfahren, basierend auf Cookies) aufgrund der Wiedererkennungsrate zu vergleichen. Des Weiteren wurde festgestellt, dass die Entscheidung, das Cosinus-Verfahren beim Webportal, von welchem die Daten gewonnen wurden, einzusetzen, die optimale Entscheidung war. Unter allen sieben getesteten Proximitätsmaßen hat dieses am besten abgeschnitten. Zu beachten ist, dass diese Aussage nicht automatisch auf andere Webauftritte übertragen werden kann. Die Gewichtung aufgrund der Transinformation in Kombination mit dem Cosinus-Koeffizienten verschlechterte das Ergebnis der Wiedererkennung.

Abschließend kann festgehalten werden, dass mit dieser Arbeit eine solide Basis für weitere Untersuchungen geschaffen wurde.

Anhang A: Abbildungen

Name	Typ	Row-Key	Column Name	Column Wert	Beschreibung
Attribute	normal	Name bzw. Typ des Attributes[1]	Wert des Attributes	gemappte ID (Integer)	Enthält sämtliche Attributausprägungen und dient als Lookup der gemappten ID.
Client	normal	Time UUID	lastTime	Zeitstempel des neuesten Visit	Ordnet jedem Client den Zeitstempel des neuesten Visits zu.
ClientToTag	normal	Time UUID (Client)	Name des Tags	Zeitstempel	Enthält für jeden Client die Liste mit allen Tags und dem jeweils letztmaligen Auftreten eines Tags.
ClientToVisit	normal	Time UUID (Client)	Key des Visit	Zeitstempel der Besuchszeit	Enthält für jeden Visit die Liste mit allen zugeordneten Visits und dem Zeitstempel des Auftretens.
IdToVisit	normal	Id (Integer)	Key des Visit	Stringliste der Attribut-Ids	Enthält zu jeder ID sämtliche Visits, die identisch sind, und die Liste mit den Attribut-IDs.
Tag	normal	Tagname	Key des Client	Zeitstempel des neuesten Visit	Listet zu jedem Tag die zugehörigen Visits und den Zeitstempel auf.
Visit	Super	Time UUID	Super: properties, Attributtyp Inner: Name bzw. Typ	Wert des Attributes oder der Eigenschaft	Dient zur Ablage der Visits mit allen Attributen und zusätzlichen Merkmalen eines Visits.

Abbildung A.1: Column Families des Fingerprint-Service

Anhang B: Tabellen

Attribut	Anzahl Ausprägungen	Anzahl Auftritte
useragent	22919	507204
accept-language	1671	506369
video	1447	507113
fantasy	715	482771
serif	553	507084
Courier	341	501990
Symbol	335	487976
Script	325	314205
Roman	294	304877
Wingdings	287	467303
Modern	287	304734
cursive	279	496222
Arial CYR	259	256376
Century Gothic	251	410311
Wide Latin	247	291699
Bookman Old Style	245	407079
Dingbats	244	57313
Helvetica	244	457903
Flash	231	474551
Liberation Mono	229	57523
Times New Roman Cyr	229	91774
Cambria	228	422687
DejaVu Sans Mono	227	124533
FreeSans	227	57055
URW Gothic L	223	56939
Antiqua	222	55839
Avalon	222	53574
Century Schoolbook L	222	56950
Frosty	220	56471
Nimbus Sans L	220	56949
Bitstream Charter	217	56868
Verona	217	53771
Garuda	215	56292
Gentium	215	56028
Droid Sans Mono	213	54666
Fraktur	213	53171
Droid Serif	212	54696
Minion Web	210	53404
Blackletter	209	53189

Tabelle B.1: Auflistung der Attribute und deren Eigenschaften: Teil 1

Attribut	Anzahl Ausprägungen	Anzahl Auftritte
King	208	53143
Decorative	206	53097
TeX	206	53099
Palatino	205	100016
Trebuchet MS	205	500850
Lalit	204	53095
Arial Narrow	202	408578
Java	202	286264
SimSun	201	308451
AR PL SungtiL GB	198	51596
Arial	195	500570
Calibri	195	424466
Constantia	193	422666
Comic Sans MS	169	486056
Adobe Reader	124	300683
Times	123	97179
Windows Media Player	105	205058
accept-charset	86	264142
timezone	79	507113
Shockwave	62	92461
accept	61	507213
Quicktime	53	189953
VLC	51	19806
DivX	45	34414
accept-encoding	43	494114
Silverlight	42	255587
DevalVR	20	451
Totem	20	2950
flash	9	17
silverlight	7	12
adobe reader	6	11
cambria	5	18
courier	5	18
modern	5	10
roman	5	10
script	5	10
symbol	5	18
wingdings	5	18
arial	4	18

Tabelle B.2: Auflistung der Attribute und deren Eigenschaften: Teil 2

Attribut	Anzahl Ausprägungen	Anzahl Auftritte
arial cyr	4	10
arial narrow	4	12
bookman old style	4	12
calibri	4	18
century gothic	4	11
comic sans ms	4	18
constantia	4	18
helvetica	4	18
simsun	4	13
trebuchet ms	4	18
wide latin	4	11
windows media player	4	14
dejavu sans mono	3	6
java	3	6
quicktime	3	5
shockwave	3	3
times	3	7
times new roman cyr	3	4
antiqua	2	3
ar pl sungtil gb	2	3
avalon	2	3
bitstream charter	2	3
blackletter	2	3
century schoolbook l	2	3
decorative	2	3
dingbats	2	3
droid sans mono	2	3
droid serif	2	3
fraktur	2	3
freesans	2	3
frosty	2	3
garuda	2	3
gentium	2	3
king	2	3
lalit	2	3
liberation mono	2	3
minion web	2	3
nimbus sans l	2	3
palatino	2	3
tex	2	3
undefined	2	1
urw gothic l	2	3
verona	2	3

Tabelle B.3: Auflistung der Attribute und deren Eigenschaften: Teil 3

Literaturverzeichnis

[1] Cassandra wiki. http://wiki.apache.org/cassandra/, 2012. [Online; Stand 18. August 2012].

[2] Rapidminer. http://rapid-i.com/content/view/181/190/, 2012. [Online; Stand 18. August 2012].

[3] Christian Breiteneder. Distanz und Ähnlichkeit. http://www.ims.tuwien.ac.at/teaching/mm2/MM2-Teil6-Distanz.pdf, 2012. [Online; Stand 18. August 2012].

[4] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications and Signal Processing. John Wiley & Sons, 2006.

[5] Peter Eckersley. How unique is your web browser? https://panopticlick.eff.org/browser-uniqueness.pdf, 2010. [Online; Stand 18. August 2012].

[6] G. Fischer. *Lineare Algebra: Eine Einführung für Studienanfänger*. Grundkurs Mathematik. Vieweg+Teubner Verlag, 2009.

[7] Eric Gerds. Browser plugin detection with plugindetect. http://www.pinlady.net/PluginDetect/. [Online; Stand 22. Juli 2012].

[8] Karin Haenelt. Ähnlichkeitsmaße für vektoren. http://kontext.fraunhofer.de/haenelt/kurs/folien/Haenelt_VektorAehnlichkeit.pdf, 2004. [Online; Stand 18. August 2012].

[9] Gabriel Handford. font-detect-js. https://github.com/gabriel/font-detect-js. [Online; Stand 22. Juli 2012].

[10] Samy Kamkar. evercookie – never forget. http://samy.pl/evercookie/, 2010. [Online; Stand 18. August 2012].

[11] Teknomo Kardi. Tutorial on decision tree. http://people.revoledu.com/kardi/tutorial/DecisionTree/, 2009. [Online; Stand 18. August 2012].

[12] Steffen Mölleken. Analyse des nutzerverhaltens. http://www.kis-sm.de/onlinewerbung, 2011. [Online; Stand 27. Juli 2012].

[13] Helfried Moosbrugger. Kodierungsverfahren im allgemeinen linearen mo-

dell. http://user.uni-frankfurt.de/~moosbrug/lehre/kap0506/KODIER.pdf, 2006. [Online; Stand 18. August 2012].

[14] T.A. Runkler. *Data Mining: Methoden und Algorithmen intelligenter Datenanalyse*. Computational Intelligence. Vieweg+Teubner Verlag, 2009.

[15] Fabian Seifert. Online-fraud-detection als webservice, 2011.

[16] C. E. SHANNON. A mathematical theory of communication. http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf, 1948. [Online; Stand 18. August 2012].

[17] Tzeng ShengLi, Wu Han-Ming, and Chen Chun-Houh. Selection of proximity measures for matrix visualization of binary data. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5305137, 2009. [Online; Stand 18. August 2012].

[18] M. Werner. *Information und Codierung: Grundlagen und Anwendungen*. UniScript. Vieweg, 2002.

[19] Wikipedia. Ähnlichkeitsanalyse — wikipedia, die freie enzyklopädie. http://de.wikipedia.org/w/index.php?title=%C3%84hnlichkeitsanalyse&oldid=102260087, 2012. [Online; Stand 18. August 2012].

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, 31. August 2012